INTEGRATING ONTOLOGIES AND LARGE LANGUAGE MODELS FOR SCIENTOMETRIC QUESTION ANSWERING: A CASE STUDY IN CONTROL THEORY

D. A. Gubanov* and V. A. Sergeev**

*,**Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

*✉ dmitry.a.g@gmail.com, **✉ sergeev.bureau@gmail.com

Abstract. This paper considers the problem of automatic answering complex scientometric questions formulated in natural language (NL) over knowledge bases. The study is topical due to the limitations of modern large language models (LLMs): despite high understanding capabilities, they tend to generate inaccurate responses to user questions and may have outdated information in specialized subject areas. At the same time, knowledge graphs provide accurate and relevant information but require knowledge of a formal query language. A hybrid architecture-based solution is proposed: an LLM acts as an intelligent interface to an ontology-driven knowledge base, converting NL questions into correct SPARQL queries, and the results are returned to the user. To solve the problem, a specialized data corpus is compiled to train and test NL-to-SPARQL models in the field of control theory. The approach is implemented based on the ontology of scientific activity in the field of control theory and validated on the generated corpus of questions. Integration of the LLM with the ontology-driven knowledge base ensures a high accuracy of answers (about 99%), which confirms the prospects of this approach.

Keywords: scientometrics, graph knowledge bases, large language model (LLM).

INTRODUCTION

In recent years, large language models (LLMs) have demonstrated impressive results in natural language (NL) processing, thereby opening up new possibilities for intelligent database search; for example, see [1–5]. The ability of transformer architectures to capture the semantic context of a query has enabled passing from keyword-based to meaning-based search; however, the application of such models in specialized subject areas is fraught with serious limitations. In tasks requiring accuracy and relevance (such as scientometric analysis), the models are prone to "hallucinate" (generate plausible but false answers) and unable to operate fresh data appeared after the completion of their training [6–8].

An alternative approach involves structured knowledge bases and Linked Data technologies. Ontologies provide a rigorous interpretation of terms, while knowledge graphs (KGs) serve as a reliable source of facts. Knowledge graph data are accessed using formal query languages such as SPARQL. This search method guarantees the interpretability and reproducibility of results, which is critically important for the analysis of scientific activity. However, the use of SPARQL creates a high barrier to entry: a correct query requires specialized knowledge and skills, and existing graphical interfaces limit the user to a set of standard filters.

The automatic translation of a natural language into formal queries (NL-to-SPARQL) aims to eliminate this barrier. Despite considerable progress in this field [9], existing solutions face challenges when processing complex analytical queries. Unlike simple "factoid" questions (e.g., *"Who is the author of paper X?*"), analytical questions in scientometrics often require multistage processing, aggregation, comparison of quantitative indicators, and filtering based on complex conditions (e.g., *"Find the most active authors on topic A* (*over the past 5 years*) *affiliated with organization B*

who have, directly or indirectly (*through a chain*), *cited researcher C*"). Standard approaches often generate syntactically incorrect code or make mistakes in the logic of relations between entities.

Scientific analytics and scientometrics are applied fields that can particularly benefit from combining LLMs with semantic technologies. The volume of scientific information—publications, researcher data, citations, etc.—is steadily growing, and intelligent analysis methods are required to make sense of it. Linked Data technologies provide tools for integrating heterogeneous information about scientific activity into a unified knowledge graph with formalized concepts (e.g., researcher, paper, journal, organization, research topic) and relations between them (authorship, citation, organizational affiliation, topic development, etc.). Large open knowledge graphs of this kind already exist, such as the Microsoft Academic Knowledge Graph (MAKG) or SemOpenAlex (the former's successor), which contain billions of triples about publications, authors, and organizations [10, 11].

This paper proposes a solution to the above problem, using the field of control theory as an example. We present an approach to integrating LLMs with the ISAND (Information System for Scientific Activity Analysis) ontology-driven knowledge base [12]. Here, an LLM is used as an intelligent interface that converts NL questions into valid SPARQL queries to a knowledge base in a particular scientific field.

The scientific contribution of this work is as follows. An architecture for a hybrid question answering (QA) system is developed and adapted to process complex analytical queries in highly specialized scientific fields, unlike existing approaches focused on open and broad subject areas (Wikidata) and factoid questions. A Russian-language dataset on control theory is created and described to train NL-to-SPARQL models in specialized subject areas. (Such datasets are currently in deficit.) The effectiveness of fine-tuning an LLM for generating SPARQL code for a complex data schema is experimentally confirmed.

The remainder of this paper is organized as follows. Section 1 presents the methods used and the architecture of the hybrid system. In Section 2, we describe the data—the ontology-driven knowledge base and the generated set of queries. Section 3 is devoted to experimental results and their analysis. Finally, the findings of this study and some directions for future work are outlined in the Conclusions.

1. METHODS

Early research on converting NL questions into SPARQL queries primarily involved semantic parsing methods and used templates with rules. Such approaches demonstrate acceptable quality when handling simple, straightforward queries, but suffer from significant limitations in processing complex queries with nested structures or aggregate functions. Early systems, such as PowerAqua [13] or Aqqu [14], used sets of templates to map NL triples into ontological triples in knowledge graphs, which considerably restricted their application to questions with simple structures. Following the emergence of large corpora of "NL question–SPARQL query" pairs, such as QALD (*Question Answering over Linked Data*) [9], it became possible to apply neural network models. The generation of SPARQL queries was treated as a machine translation task, in which a question formulated in a natural language serves as the source text, and the SPARQL query as the target text. For example, an approach based on recurrent neural networks with the LSTM (*Long Short-Term Memory*) architecture was proposed in [15]. However, despite significant progress, processing complex multi-component queries still poses a challenge.

Currently, the integration of LLMs with graph knowledge bases is considered a promising line for building advanced QA systems. There are several main schemes for integrating LLMs with ontologies of scientific activity based on graph databases.

1. **An LLM as an interface to a KG**. The model generates a SPARQL query (or another formal query) based on the user's question; this query is then executed on the knowledge graph, and the result is returned to the user. Hiding the query language's complexity from the user, this approach, however, requires the generated LLM query to be correct; for example, see [16].

2. **A KG as a knowledge source for an LLM**. Before generating a response, the model extracts relevant facts from the graph (e.g., finds related triples by key entities in the question) and incorporates them into its input context. This approach implements the RAG (*Retrieval-augmented generation*) strategy: an external knowledge base is used to generate a prompt, thereby improving the factual accuracy of the response (for example, see [17, 18]).

3. **A KG for verifying and refining the responses of an LLM**. The model first generates a draft response

or a list of hypotheses (possibly with intermediate reasoning), after which the knowledge base is activated: the system verifies and refines the results, e.g., by filtering or ranking the generated answers based on their confirmability by triples, or by correcting factual errors (for example, see [19]).

The approaches mentioned in items 2 and 3 can be performed iteratively: the model sequentially sends queries to the knowledge base (e.g., via a series of SPARQL queries) in the CoT (*chain-of-thought*) spirit, gradually gathering the information necessary for the answer. According to the global trend, modern question answering systems are increasingly built on such hybrid architectures (an LLM + a knowledge graph), capable of responding to complex queries while maintaining a highly natural dialog with the user.

An LLM must be trained to correctly convert NL questions into SPARQL queries (while avoiding schema hallucinations, such as inventing non-existent predicates). This can be done by prompts during inference (*in-context learning*) or by fine-tuning the model before inference. The choice in favor of fine-tuning is driven by three factors. First, the data schema is complex: a complete description of the ontology in a prompt would exhaust the context of a local model. Fine-tuning allows embedding knowledge about the graph structure in the model's weights. Second, the syntax is reliable. Fine-tuning significantly reduces the probability of generating incorrect SPARQL code compared to prompt engineering methods. Third, inference is effective. Using a smaller fine-tuned model (e.g., 7B/8B) in a real system is more cost-effective and computationally efficient than using massive prompts with examples in every query.

Fine-tuning a modern LLM may require significant computational resources, so it seems appropriate to use PEFT (*Parameter-Efficient Fine-Tuning*) methods, reducing hardware requirements for computations; for example, see [20]. In particular, a reasonable choice is to use the LoRA (*Low-Rank Adaptation*) and QLoRA (*Quantized Low-Rank Adaptation*) methods [21, 22]. Compared to LoRA, QLoRA yields an even greater reduction in GPU memory consumption, which is important for very large models. With these methods, extra training layers with fewer parameters (compared to the layers of the original model) are added to a pre-trained model. The added layers are functionally separated into a distinct block called an adapter. Adapters are trained for a specific task without changing the weights of the original model. In [21, 22], training and operation with the adapters were carried out on two A30 GPUs with a total memory of 48 GB under the openSUSE Leap 15.6 operating system.

In this study, we employ the first scheme for integrating LLMs with ontologies of scientific activity based on graph databases. The system architecture is shown in Fig. 1. The main functional blocks of the system are the LLM and KG blocks. This system operates as follows. The user formulates a query to the system, and based on this query, the system generates a prompt for the LLM. Next, the LLM converts the prompt into a SPARQL sequence. During the processing of the SPARQL sequence, the redundant code generated by the LLM is stripped, auxiliary lines are commented out, and namespace prefixes are added. Then the query execution module sends the SPARQL sequence to the database and handles network errors. Finally, using the resulting SPARQL sequence, the requested information is extracted in the JSON format from the knowledge graph and returned to the user after processing.

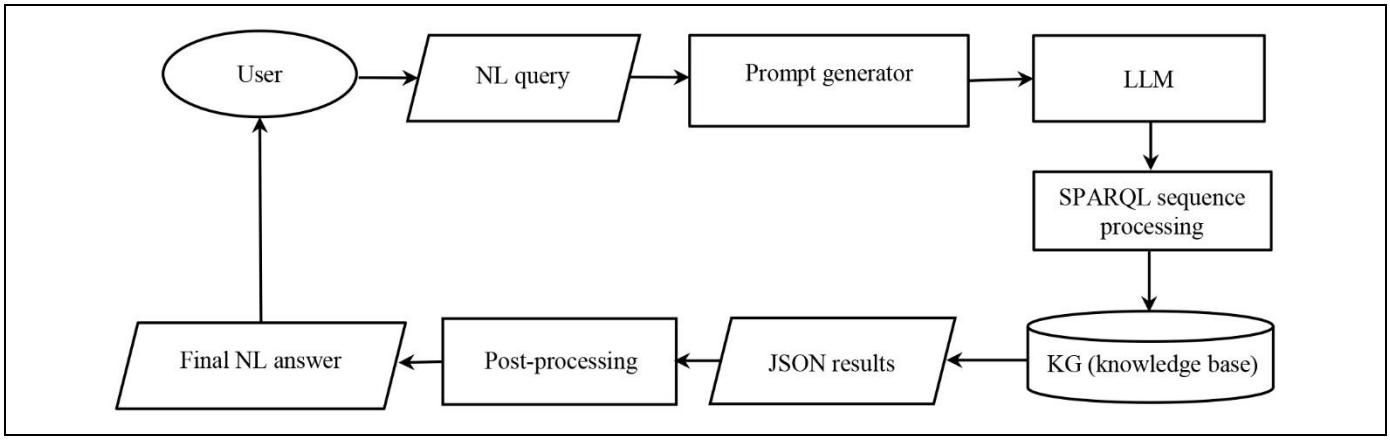

Fig. 1. The system architecture.

2. DATA

We created a specialized dataset consisting of question–answer pairs on control theory in the Russian language, including real information queries from the expert community. Unlike existing benchmarks, this dataset focuses on the field of control theory and its applications and contains high-quality annotated question–answer pairs. Each pair consists of a user's question to the system and a response in the form of a SPARQL query that is valid according to the ISAND ontology. All pairs were verified by experts. This data collection approach ensures high reliability and relevance of the test examples, also reflecting the nuances of the Russian-language terminology in control theory (which is often neglected in global English-language datasets). When compiling the dataset, experts initially created 126 question–answer pairs valid in terms of extracting information from the knowledge graph.

The total number of examples includes both 16 pairs with general-purpose questions (such as *"Выведи все организации вместе с их названиями";* in English: *"List all organizations along with their names*") and 110 pairs with questions referring to entities from the knowledge graph. The system is designed to work both with a GUI (*Graphical User Interface*), oriented toward typical scenarios, and with user questions via a text input window. Therefore, the dataset contains examples with entity instances and with IRIs (*Internationalized Resource Identifiers*) from the ISAND ontology (see the table below). Based on the aforementioned 110 pairs, question–answer pairs were automatically generated by substituting particular entity instances into the question and the SPARQL query. As a result, 1000 question–answer pairs with IRIs were generated from the graph database, and 1080 question–answer pairs with entity names (*Name*) were randomly selected from the list of typical entities found in the identifier slots.

The resulting list of 2080 question–answer pairs was supplemented by 16 pairs with general questions, leading to 2096 question–answer pairs in the final dataset. The final list was split into training (1680 pairs), validation (208 pairs), and test (208 pairs) sets. The distribution of questions by topic in the validation and test sets is the same as in the training set.

The following prompt template was used during training:
"Task: Generate SPARQL queries to query the knowledge graph based on the provided schema definition. ### Question {} ### Answer: {}."

According to the length analysis of the prompts generated, the maximum length for all tokenizers considered (see the Appendix) is limited to 512 tokens. Figure 2 shows the distribution of prompt lengths based on the generated dataset, using the *Qwen/Qwen2.5-7B-Instruct* tokenizer as an example.

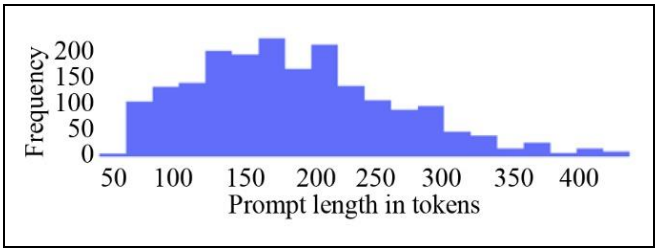

Fig. 2. The distribution of prompt lengths based on the generated dataset for the *Qwen/Qwen2.5-7B-Instruct* tokenizer.

Examples of question–answer pairs

Question	Answer	Question	Answer
Кто является автором публикации :publ_7808? In English: *Who is the author of publication :publ_7808?*	SELECT ?author WHERE { :publ_7808 :имеетАвтора ?author . }	Кто является автором публикации «Оптимизация динамических систем с импульсными управлениями и ударными воздействиями»? In English: *Who is the author of the publication "Optimization of Dynamic Systems with Pulse Controls and Shock Impacts"?*	SELECT ?author WHERE { ?publ a :Публикация; :название ?publ_name; :имеетАвтора ?author . FILTER(lcase(str(?publ_name)) = «Оптимизация динамических систем с импульсными управлениями и ударными воздействиями») }
(a) the question–answer pair with IRI :publ_7808		(b) the question–answer pair with the particular entity instance "Оптимизация динамических систем с импульсными управлениями и ударными воздействиями."	

3. RESULTS

A PEFT approach was used to train the LLM. Within this approach, two methods were considered: LoRA and QLoRA. A four-bit quantization for QLoRA adapters was obtained using Python's library *bitsandbytes*. Thus, only a share of the weights added (and not the LLM) was fine-tuned; they were functionally isolated into the so-called adapter [21].

Several leading models, suitable for question answering and deployment on two A30 GPUs from the Hugging Face leaderboard [23], were considered to solve the NL-to-SPARQL conversion problem. Academic research often faces the challenge of insufficient computing power. Therefore, we selected models with relatively low GPU memory requirements that can serve as an NL-to-SPARQL converter (e.g., *phi-3-mini-4k-i* with 4 billion parameters). To further reduce GPU memory requirements, *paged_Adam_32bit* was applied as an optimizer in LoRA models. The 4-bit adapters were trained using the *Adam_torch* optimizer. Other training settings were the same for all models under study.

The exact-match (EM) metric was taken to assess the performance of all adapters. Line break characters and duplicate spaces were removed from the LLM's response. With the EM approach, the response generated by the LLM must match the reference response character-wise. This approach eliminates the need to interpret the SPARQL query generated by the LLM for correct execution and to perform error analysis in post-processing.

Figure 3 shows the results of the LoRA and QLoRA adapters on the test dataset. The results of the best adapters are provided in the Appendix. According to the results, it is impossible to identify one model outperforming the others across all parameters considered: there exists an obvious contradiction between quality, speed, and GPU memory requirements. As practical recommendations, we suggest using the fastest models, such as *Vicuna7b-v1.5-16k (lora)* or *phi-3-mini-4k-i (lora)*, in server applications sensitive to response speed. Models with high quality scores, such as *gemma-2-9b-it 4bit*, are best suited to situations with the highest priority of reliable response. The least resource-intensive models, such as *phi-3-mini-4k-i (4bit)*, can be potentially used in mobile applications.

A general trend is that 4-bit adapters consume less GPU memory but perform more slowly. Several models achieved a quality assessment above 99% on the test dataset, and the quantized *Google/gemma-2-9b-it* successfully answered all questions in the test dataset. Also, note that the adapter based on *phi-3-mini-4k-i*

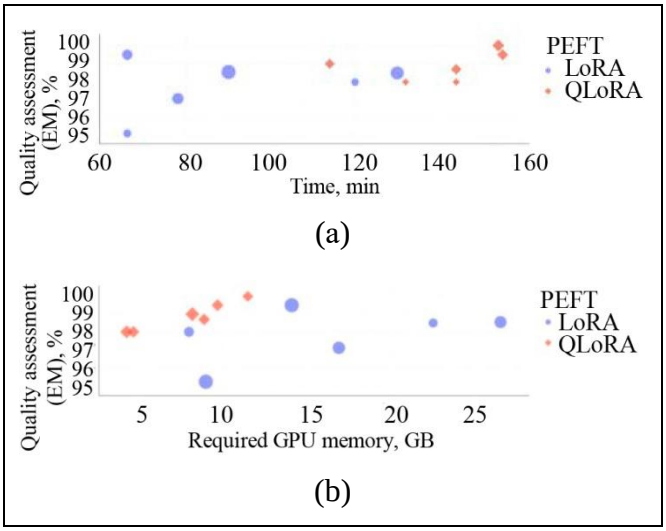

Fig. 3. The performance of the models under consideration (each point corresponds to a single model): (a) the size of the shape reflects the allocated amount of GPU memory; (b) the size of the shape reflects the execution time.

with 4 billion parameters (a relatively small model) performed well on all test questions, using only 4.4 GB of GPU memory.

Figure 4 shows the QA user interface implementing the NL→SPARQL chain for the ontology-driven knowledge graph. The text box at the top is intended to enter a natural language query (e.g., "*Кто является автором публикации Матричная теорема о лесах и лапласовские матрицы орграфов?*"; in English: "*Who is the author of the publication The Matrix Theorem on Forests and Laplacian Matrices of Digraphs?*"); for each query, a SPARQL query is automatically generated by pressing the *Generate* button. The resulting code is displayed in the *SPARQL Query* Editor with line numbering and syntax highlighting. This block includes standard prefixes (rdfs, xsd), the ontology's domain prefix (https://www.ipu.ru/ontologies/isand#), as well as the query body built on the ontology's classes and properties (e.g., :Публикация, :имеетАвтора, :годИздания, :фамилия, :имя; in English: *:Publication, :has Author, :yearofIssue, :last name, :first name*). The SPARQL query execution block is located below: the *Run SPARQL* button sends the query to the SPARQL endpoint of the graph database, while the *Answer (JSON)* panel returns a machine-readable result in the JSON format to simplify subsequent processing and verification. The *Save to history* button places the current query in the history log. The *History* table records the queries, with the possibility of editing and deletion. Thus, the interface simultaneously supports interactive SPARQL generation, transparent execution with controlled output, and query history.

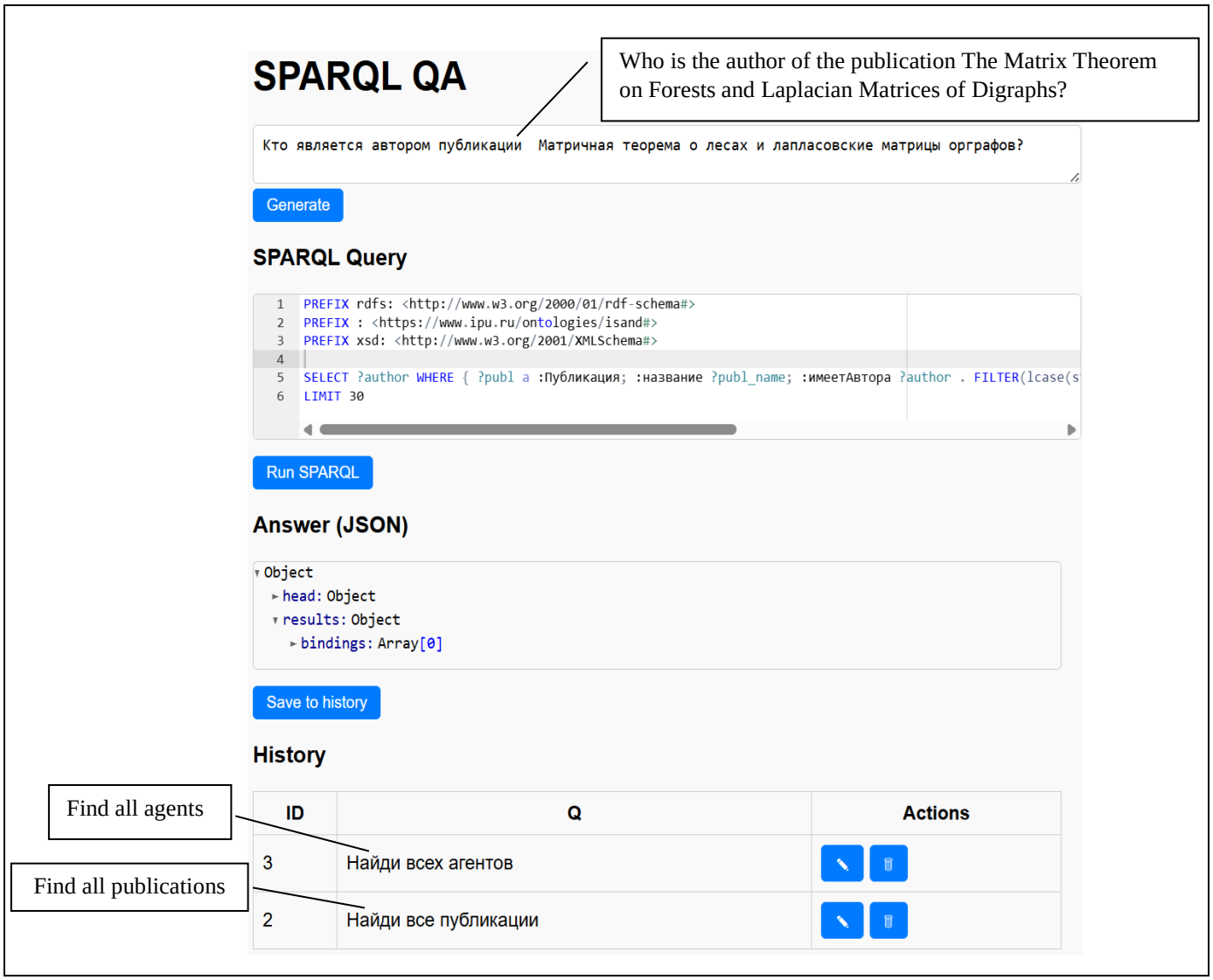

Fig. 4. The user interface for executing queries.

According to the results, the developed system can successfully handle user queries, demonstrating the advantages of combining an LLM with a graph knowledge base.

CONCLUSIONS

A hybrid question answering system has been developed by integrating the capabilities of large language models (LLMs) and a graph knowledge base to execute complex scientometric queries. This approach overcomes several limitations of conventional methods: the system automatically converts a natural-language question into one or more SPARQL queries to the knowledge base instead of manually composing formal queries or using rigid templates. The high effectiveness of using LLMs as an interface to knowledge graphs has been confirmed within the approach. The possibility of quickly and resource-efficiently adapting LLMs to a particular ontology has been demonstrated.

A promising step is to verify the solution's scalability and integrate it with larger and more complex knowledge graphs, such as *SemOpenAlex*. It is also of interest to improve the system by utilizing text knowledge sources, primarily scientific publications, to supplement, refine, and contextually justify the answers generated by the system. Finally, as the ontology-driven knowledge base is filled with current information, an important task is to update the model's knowledge accordingly.

APPENDIX

The results demonstrated by the models' adapters on a test dataset

Model	Quality (EM), %	Runtime (208 examples), min	GPU memory required, GB
Vicuna 13b-v1.5-16k LORA	98.55	90	26.5
Vicuna 13b-v1.5-16k 4-bit	99.5	155	9.7
Vicuna 7b-v1.5-16k lora	99.5	66	14.1
Vicuna 7b-v1.5-16k 4-bit	99	114	8.2
Qwen2.5-7b-instruct lora	97.1	78	16.9
Qwen2.5-7b-instruct 4bit	98.7	144	8.9
Qwen2.5-3b-instruct lora	98	120	8
Qwen2.5-3b-instruct 4bit	98	144	4.7
Gemma-2-9b-it lora	98.5	130	22.5
Gemma-2-9b-it 4-bit	100	154	11.5
Phi-3-mini-4k-i lora	95.19	66	9
Phi-3-mini-4k-i 4-bit	98	132	4.3

REFERENCES

1. Wei, J., Wang, X., Schuurmans, D., et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, *arXiv:2201.11903*, 2022. DOI: https://doi.org/10.48550/arXiv.2201.11903
2. Touvron, H., Lavril, T., Izacard, G., et al., LLaMA: Open and Efficient Foundation Language Models, *arXiv:2302.13971*, 2023. DOI: https://doi.org/10.48550/arXiv.2302.13971
3. Yang, A., Li, A., Yang, B., et al., Qwen3 Technical Report, *arXiv:2505.09388*, 2025. DOI: https://doi.org/10.48550/arXiv.2505.09388
4. Sharma, S., Tuli, S., and Badam, N., Challenges and Applications of Large Language Models: A Comparison of GPT and DeepSeek Family of Models, *arXiv.2307.10169*, 2023. DOI: https://doi.org/10.48550/arXiv.2307.10169
5. OpenAI GPT-4 Technical Report, *arXiv:2303.08774*, 2023. DOI: https://doi.org/10.48550/arXiv.2303.08774
6. Sima, A.-C. and Farias, T.M., On the Potential of Artificial Intelligence Chatbots for Data Exploration of Federated Bioinformatics Knowledge Graphs, *arXiv:2304.10427*, 2023. DOI: https://doi.org/10.48550/arXiv.2304.10427
7. Prabhune, S. and Berndt, D.J. Deploying Large Language Models with Retrieval Augmented Generation, *arXiv:2411.11895*, 2024. DOI: https://doi.org/10.48550/arXiv.2411.11895
8. Klager, G.G. and Polleres, A., Is GPT Fit for KGQA?, *CEUR Workshop Proceedings*, 2023, vol. 3447, art. no. 11. (Proceedings of the 2nd International Workshop on Knowledge Graph Generation from Text (TEXT2KG 2023).)
9. Lopez, V., Unger, C., and Cimiano, P., Evaluating Question Answering over Linked Data, *Journal of Web Semantics*, 2013, vol. 21, pp. 3–13.
10. Färber, M., The Microsoft Academic Knowledge Graph: A Linked Data Source with 8 Billion Triples of Scholarly Data, in *Lecture Notes in Computer Science*, Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., and Gandon, F., Eds., Cham: Springer International Publishing, 2019, vol. 11778, pp. 113–129. (Proceedings of the 18th International Semantic Web Conference.)
11. Färber, M., Lamprecht, D., Krause, J., et al., SemOpenAlex: The Scientific Landscape in 26 Billion RDF Triples, *arXiv:2308.03671*, 2023. DOI: https://doi.org/10.48550/arXiv.2308.03671
12. Gubanov, D.A., Kuznetsov, O.P, Kurako, E.A., et al., ISAND: An Information System for Scientific Activity Analysis (in the Field of Control Theory and Its Applications), *Control Sciences*, 2024, no. 3, pp. 35–55.
13. Lopez, V., Fernández, M., Stieler, N., and Mooa, E., PowerAqua: Supporting Users in Querying and Exploring the Semantic Web Content, *Semantic Web*, 2012, vol. 3, no. 3, pp. 249–265.
14. Bast, H. and Haussmann, E., More Accurate Question Answering on Freebase, *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, Melbourne, 2015, pp. 1431–1440.
15. Luz, F.F. and Finger, M., Semantic Parsing Natural Language into SPARQL: Improving Target Language Representation with Neural Attention, *arXiv:1803.04329*, 2018. DOI: https://doi.org/10.48550/arXiv.1803.04329
16. Rangel, J.C., Mendes de Farias, T., Sima, A.C., and Kobayashi, N., SPARQL Generation: An Analysis on Fine-Tuning OpenLLaMA for Question Answering over a Life Science Knowledge Graph, *arXiv:2402.04627*, 2024. DOI: https://doi.org/10.48550/arXiv.2402.04627
17. Fan, W., Ding, Y., Ning, L., et al., A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models, *arXiv:2405.06211*, 2024. DOI: https://doi.org/10.48550/arXiv.2405.06211
18. Gao, Y., Xiong, Y., Gao, X., et al., Retrieval-Augmented Generation for Large Language Models: A Survey, *arXiv:2312.10997*, 2024. DOI: https://doi.org/10.48550/arXiv.2312.10997
19. Kwan, L., Omran, P.G., and Taylor, K., Using Knowledge Graphs and Agentic LLMs for Factuality Text Assessment and Improvement, *CEUR Workshop Proceedings*, 2024, vol. 3828, art. no. 18. (Proceedings of Posters, Demos, and Industry Tracks at ISWC 2024.)
20. Lialin, V., Deshpande, V., Yao, X., and Rumshisky, A., Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-

Tuning, *arXiv:2303.15647*, 2023. DOI: 10.48550/arXiv.2303.15647
21. Hu, E.J., Shen, Y., Wallis, P., and Allen-Zhu, Z., LoRA: Low-Rank Adaptation of Large Language Models, *arXiv:2106.09685*, 2021. DOI: 10.48550/arXiv.2106.09685
22. Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L., QLoRA: Efficient Finetuning of Quantized LLMs, *arXiv:2305.14314*. 2023. DOI: 10.48550/arXiv.2305.14314
23. Fourrier, C., Habib, N., Lozovskaya, A., et al., Open LLM Leaderboard v2. URL: https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard (Accessed April 2, 2025.)

This paper was recommended for publication by O.P. Kuznetsov, a member of the Editorial Board.

Received September 23, 2025,
and revised November 23, 2025.
Accepted December 16, 2025.

Author information

Gubanov, Dmitry Alekseevich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ dmitry.a.g@gmail.com
ORCID iD: https://orcid.org/0000-0002-0099-3386

Sergeev, Vladimir Aleksandrovich. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ sergeev.bureau@gmail.com
ORCID iD: https://orcid.org/0000-0002-7948-8656

Cite this paper

Gubanov, D.A. and Sergeev, V.A., Integrating Ontologies and Large Language Models for Scientometric Question Answering: A Case Study in Control Theory. *Control Sciences* **2**, 43–50 (2026).

Original Russian Text © Gubanov, D.A., Sergeev, V.A., 2026, published in *Problemy Upravleniya*, 2026, no. 1, pp. 49–57.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com