

LLM Based Chatbot to Find Railway Topology and Rail Vehicle Information in Europe

Mohammed H. Rasheed^{1,*}, Marina Aguado^{1,†}

¹University of the Basque Country, Spain

Abstract

The Interoperable Europe Act came into effect in April this year. Semantic interoperability as a key element can ensure cross-border interoperability between various public services such as rail and transport services. This demo introduces a chatbot engine that employs Large Language Model (LLM) to facilitate human interaction with domain-specific Knowledge Graphs (KG) governed by the European Union Agency for Railways. The chatbot engine facilitates domain-specific SPARQL query generation based on natural language query, thereby providing an intuitive interface for non-expert users to retrieve in-domain knowledge. In contrast, our chatbot automated query generation allows domain experts to directly query triple data stores more intuitively and without intermediaries, contributing to improving the quality of hosted information. The chatbot engine uses a zero-shot prompting approach by taking the user's natural language query as an input and translates into SPARQL query to retrieve factual knowledge from the target KG. To improve the quality of the generated SPARQL query and thus improve the relevancy of results corresponding to the user question, the LLM is supported with in-domain knowledge by injecting extracted KG vocabulary information along with the user natural language question as an input. The experiments conducted on twenty in-domain competency questions revealed that leveraging LLM is a promising approach and can be efficiently oriented to be utilized in domain-specific KGs to increase productivity, reduce query construction time and increase usability by allowing non-technical users to obtain knowledge intuitively. Furthermore, the chatbot offers a very fine feature inherited from LLMs, which is the ability to answer multilingual queries, allowing its utilization among a wide range of users regardless of any language boundaries which ultimately contributes to the cross-border interoperability of public services in countries using different languages.

Keywords

Large Language Models, KGQA, SPARQL, RDF, Prompt Engineering

1. Introduction

Large language models (LLMs) are experiencing notable advances in their performance and functionalities; as a result, LLMs have been integrated and applied in various fields to provide both mainstream and downstream tasks [1][2][3]. The proven ability of LLM to process natural language has opened the door for integration into many applications, specifically text-based applications, including Knowledge Graphs Question Answering (KGQA) through the translation of natural language (NL) questions into SPARQL queries [4][3][5]. This demo paper and from a

EKAU 2024: EKAU 2024 Workshops, Tutorials, Posters and Demos, 24th International Conference on Knowledge Engineering and Knowledge Management (EKAU 2024), November 26-28, 2024, Amsterdam, The Netherlands.

*Corresponding author.

†These authors contributed equally.

✉ mhameed001@ikasle.ehu.eus (M. H. Rasheed); marina.aguado@ehu.eus (M. Aguado)

🆔 0000-0002-6653-5484 (M. H. Rasheed); 0000-0002-9417-7223 (M. Aguado)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

general perspective is part of an ongoing experiments to examine and evaluate the capabilities of LLM to translate NL questions to SPARQL queries over domain-specific KG, specifically the European Union Railway Agency (ERA) Knowledge Graph (KG) motivated by the fact that there are limited attempts to target such domain-specific KGs.

Constructing and composing SPARQL queries that can interrogate and retrieve responses from domain-specific KGs is a challenging task on the technical and temporal levels [6][7]. It requires extensive experience and proficiency in KG querying languages such as SPARQL, as well as a deep understanding of the architecture and vocabulary of the target KG [3][8][9]. To bridge the gap between human-KG interaction, LLM can be utilized as an interface. Such an interface can first simplify the process of querying general and domain-specific KGs, allowing users without technical proficiency in KG’s querying languages such as SPARQL to interact and retrieve information based on NL as input. Furthermore, composing SPARQL queries using NL can increase productivity, reduce complexity, and reduce the construction time of manual queries[8][3]. Lastly, LLMs can provide an intuitive and user-friendly environment that allows interaction with various KG complexity.

Based on the above, our chatbot engine provides an LLM-based interface engine that aims to facilitate information retrieval from domain-specific KGs. Our approach proposes an unadorned fashion approach to perform a zero-shot SPARQL query generation by augmenting LLMs with previously extracted KG vocabulary information. Despite its great impact on the quality of responses, the chatbot does not utilize any natural language processing (NLP) techniques over the input text, therefore concentrating solely on the capability of LLM to generate a SPARQL query based on NL user input augmented with extracted KG vocabulary information.

The rest of the paper is structured as follows: In Section 2 we present related work. Section 3 introduces the design and architecture of the chatbot engine. Section 4 demonstrates the chatbot interface and its operations. Performance and evaluation of our proposed engine are presented in Section 5. Finally, Section 6 summarizes the contribution of this demo and suggests insights for future work.

2. Literature Review

A considerable amount of studies have explored the translation of NL questions to SPARQL queries with and without LLMs. However, the expressiveness and ambiguity of natural language led to the emergence of various non-deterministic approaches and methods to perform the translation task. With recent advances in LLMs, many researches have leveraged the use of LLMs to support the translation process at various locations within the translation process pipeline. [10] designed a one-shot prompt-based template to instruct LLM to generate SPARQL from NL based on similar labeled examples extracted based on three publicly available datasets. Another promising approach with high leaderboard scores is proposed by [11] which focuses on fine-tuning LLMs using ground truth datasets to generate the SPARQL-based NL question. [12][13] utilized semantic and syntactic analysis of the NL question to identify entities and relations in the input text, then perform a URI lookup against the KG to retrieve the corresponding KG entities and relations to generate a SPARQL representation of the NL question. Another LLM based approach is proposed by [14], proposing an approach that uses few shot LLM prompts

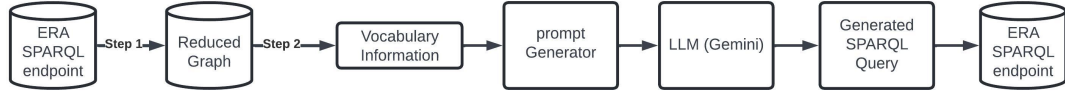


Figure 1: Overview of the Underlying Design Architecture.

with top-n labeled examples that has a context similar to the NL question to retrieve scholarly information from SciQA benchmark. Their approach is only effective when the top-n labeled examples match the input question. [5] explored the potential of using ChatGPT to support related KG tasks, among these tasks is the generation of SPARQL from NL. The experiment was based on a self-created small KG and was therefore very limited. [7] proposed using controlled natural language as a middle step to answer the NL question over KGs using semantic parsing via LLMs, addressing the unambiguous translation of controlled natural languages into SPARQL queries. In conclusion, leveraging LLMs to translate natural language into SPARQL queries has shown promising results in various domains, including semantic analysis, KGQA, and syntactic formulation. However, most of the approaches targeting public KGs supported by already available ground truth competency questions or ground truth datasets to train or fine-tune LLMs in multi-shot bases, whereas our approach uses only domain-specific vocabulary and KG information in a zero-shot prompt to support the translation of NL to SPARQL with minimal token sizes.

3. Design and Architecture

Our proposed test bed is used to explore the ability of LLM to generate SPARQL queries and retrieve information from a domain-specific KG using NL questions. As illustrated in Figure 1, our method involves generating an enhanced prompt augmented with in-domain KG ontology vocabulary information extracted from the Register of Infrastructure (RINF¹) ontology KG. The generated prompt is then fed to the LLM in the form of a zero-shot prompt[15][8] to translate the NL question into a SPARQL query, which is used to obtain the answer from the ERA SPARQL endpoint. Our test bed engine uses the Gemini API (gemini-pro) from Google². To preserve the randomness, diversity, and consistency of the responses, our test bed kept the temperature constant (value: 4) throughout all tests. The SPARQL query generated by the LLM is then executed against the ERA SPARQL endpoint to retrieve the responses back to the user.

The chatbot engine demo is implemented using Streamlit framework³ allowing users to easily interact and explore the RINF KG information using the Gemini API.

¹<https://data-interop.era.europa.eu/era-vocabulary/>

²<https://gemini.google.com/app>

³<https://streamlit.io/>

Write Your Natural Language Question: ?

What is the longest section of line and in what country Natural Language Question

Select The LLM you like to Use:

Gemini

Gemini

Generate SPARQL Query

Result of Executing sparql query against ERA endpoint Using Gemini

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?section ?country (MAX(?length) as ?max_length)
WHERE {
  ?section rdf:type <http://data.europa.eu/949/SectionOfLine> .
  ?section <http://data.europa.eu/949/inCountry> ?country .
  ?section <http://data.europa.eu/949/length> ?length .
}
GROUP BY ?section ?country
ORDER BY DESC(?max_length)
LIMIT 1

```

SPARQL Generated

Output Using Prompt1

	section	country
0	http://data.europa.eu/949/functionalInfrastructure/sectionsOfLine/04.00.00_ELO405	http://publications.euroj

Response from ERA Endpoint

Figure 2: Chatbot SPARQL Query Generation and Answer Retrieval.

4. Chatbot Engine Demonstration

The chatbot engine provides an intuitive and user-friendly interface and can be accessed through this link ⁴. The chatbot allows the user to input their own question in natural language that targets the ERA RINF Ontology KG as demonstrated in Figure 2. Alternatively, if the user is unfamiliar with the domain in question, the chatbot provides a drop-down list of suggested sample questions of varying complexity that can be used to test the tool.

After feeding the requested question into the input area and selecting Generate SPARQL Query, the tool and behind the seen will generate a zero-shot prompt, fed to Gemini API to translate the input text from unstructured NL form to a structured SPARQL query format, ultimately querying the ERA SPARQL endpoint to retrieve responses as shown in Figure 2. It is important to note that LLM inherently may not provide a consistent answer on every run; if the query fails initially, trying again may yield an answer.

Ultimately, the tool keeps track and saves all questions with the corresponding responses generated by the chatbot. The recorded information will be used to improve and fine-tune the chatbot engine for future advancements.

5. Chatbot Engine Performance

Our chatbot engine demo demonstrated promising performance in translating competency questions of varying complexity. Although some questions are straightforward and have their

⁴<https://llm-chatbot-translator.streamlit.app/>

answers available explicitly, others are complex and their answers are available implicitly and require visiting multiple triple paths to aggregate answers from KG. For example, the question: ***What is the longest section of line?*** suggests that the data in the target KG contain multiple instances of entities labeled as *"section of line"* with associated lengths. This type of question is classified as a complex question, since its corresponding answer is not explicitly available in the KG. It requires visiting more than triple paths to retrieve the required information, performing comparisons across multiple data paths, and aggregation to determine the correct answer as shown in Figure 3. Based on aforementioned, answering such complex questions by humans requires a clear and thorough understanding of the KG schema, structure of entities and associated data, length value representation, and vocabulary used to express related information. Therefore, leveraging LLM to perform all these operations can support steps towards simplifying interaction with KG-based systems and increase user productivity and usability.

6. Conclusion, Envisaged Benefit and Future Development

In this chatbot engine demonstration, we have showcased that utilizing LLMs to perform KGQA by translating NL to SPARQL query to retrieve information from domain-specific KG is promising and effective. Undoubtedly, leveraging LLM in KGQA systems can increase productivity, reduce query construction time, and usability by allowing non-technical users to obtain answers from any KG. As a next step, this chatbot can be transformed into a reusable component to facilitate the query of domain-specific triple stores. It will also allow an automatic benchmarking of different LLMs, since tests can be easily automated and tested against different LLM models. Another feature is to capture domain-specific competency questions for further analysis. We also expect to identify and find key performance indicators to define the LLM requirements to serve adequately as an end-user interface. In addition to the aforementioned functionality and features, the chatbot offers a very fine feature inherited from LLMs, which is the ability to answer multilingual queries, allowing a wide range of users to utilize regardless of any language boundaries. In other words, the core ontology does not need to be multilingual in order to allow multilingual queries, as LLM will take this translation burden.

References

- [1] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, M. Du, Explainability for large language models: A survey, *ACM Transactions on Intelligent Systems and Technology* 15 (2024) 1–38.
- [2] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, *ACM Transactions on Intelligent Systems and Technology* 15 (2024) 1–45.
- [3] D. Myers, R. Mohawesh, V. I. Chellaboina, A. L. Sathvik, P. Venkatesh, Y.-H. Ho, H. Henshaw, M. Alhawawreh, D. Berdik, Y. Jararweh, Foundation and large language models: fundamentals, challenges, opportunities, and social impacts, *Cluster Computing* 27 (2024).
- [4] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz,

- D. Roth, Recent advances in natural language processing via large pre-trained language models: A survey, *ACM Computing Surveys* 56 (2023) 1–40.
- [5] L. Meyer, C. Stadler, J. Frey, N. Radtke, K. Junghanns, R. Meissner, G. Dziwis, K. Bulert, M. Martin, Llm-assisted knowledge graph engineering: experiments with chatgpt (2023), in: conference proceedings of AI-Tomorrow-23, volume 29, 2023, pp. 6–2023.
 - [6] J. Liu, B. Mozafari, Query rewriting via large language models, *arXiv preprint arXiv:2403.09060* (2024).
 - [7] J. Lehmann, P. Gattogi, D. Bhandiwad, S. Ferré, S. Vahdati, Language models as controlled natural language semantic parsers for knowledge graph question answering, in: *European Conference on Artificial Intelligence (ECAI)*, volume 372, IOS Press, 2023, pp. 1348–1356.
 - [8] T. Abedissa Taffa, R. Usbeck, Leveraging llms in scholarly knowledge graph question answering, *arXiv e-prints* (2023) arXiv–2311.
 - [9] M. Yani, A. A. Krisnadhi, Challenges, techniques, and trends of simple knowledge graph question answering: A survey, *Information* 12 (2021). URL: <https://www.mdpi.com/2078-2489/12/7/271>. doi:10.3390/info12070271.
 - [10] L. Kovriguina, R. Teucher, D. Radyush, D. Mouromtsev, Sparqlgen: One-shot prompt-based approach for sparql query generation (2023).
 - [11] M. R. A. H. Rony, U. Kumar, R. Teucher, L. Kovriguina, J. Lehmann, Sgpt: a generative approach for sparql query generation from natural language questions, *IEEE Access* 10 (2022) 70712–70723.
 - [12] J.-D. Kim, K. B. Cohen, Natural language query processing for sparql generation: A prototype system for snomed ct, in: *Proceedings of biolink*, volume 32, 2013, p. 38.
 - [13] N. Mihindukulasooriya, G. Rossiello, P. Kapanipathi, I. Abdelaziz, S. Ravishankar, M. Yu, A. Gliozzo, S. Roukos, A. Gray, Leveraging semantic parsing for relation linking over knowledge bases, in: *International Semantic Web Conference*, Springer, 2020, pp. 402–419.
 - [14] T. A. Taffa, R. Usbeck, Leveraging llms in scholarly knowledge graph question answering, *arXiv preprint arXiv:2311.09841* (2023).
 - [15] S. Schulhoff, M. Ilie, N. Balepur, K. Kahadze, A. Liu, C. Si, Y. Li, A. Gupta, H. Han, S. Schulhoff, et al., The prompt report: A systematic survey of prompting techniques, *arXiv preprint arXiv:2406.06608* (2024).