Exploring the Underlying Questions of SPARQL Queries with LLMs

Guangyuan Piao, Pournima Sonawane, Shraddha Gupta, and Aidan OMahony

Dell Technologies, Ireland, {firstname.lastname}@dell.com

Abstract. SPARQL queries play a crucial role in exploring knowledge graphs (KGs) and have been widely used in practice. However, understanding what questions are actually asked to KGs by exploring queries directly is a daunting task. In line with recent efforts to leverage Large Language Models (LLMs) for deriving underlying questions of SPARQL queries, we further investigate whether increasing the number of examples in prompting and Chain-of-Thought prompting can improve the performance. Additionally, we examine whether a fine-tuned LLM with one dataset can be used on another dataset to further improve performance[1].

Keywords: SPARQL Query · Large Language Models.

1 Introduction

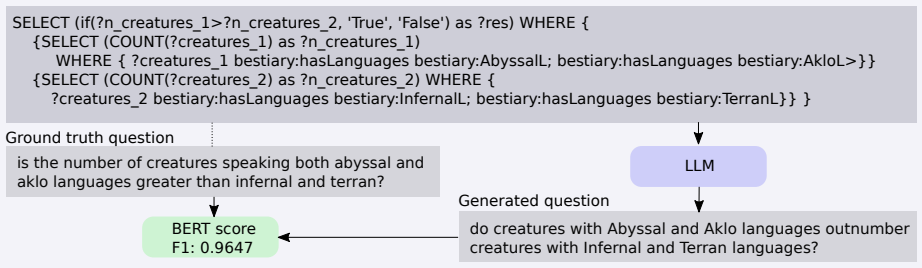
SPARQL has been widely used today for querying knowledge graphs. Despite its usefulness, exploring SPARQL queries themselves is challenging not only for lay users but also for those with substantial domain knowledge (see Fig. 1) when trying to understand the corresponding question of a SPARQL query. To tackle this challenge, SPARQL2NL [2] introduced the task of translating SPARQL into natural language to make it easier for humans to understand. However, their primary focus is on verbalizing or explaining queries. More recently, LLM (Large Language Model)-based approaches have been proposed to derive the corresponding questions of SPARQL queries [4]. However, it remains unclear whether increasing the number of examples in n-shot prompting or Chain-of-Thought prompting improves the performance of translating SPARQL queries to underlying questions (SPARQL2Q). Also, can a fine-tuned LLM using one dataset from [4] improve the performance on another dataset? We shed light on these questions in this poster.

2 Question Generation for SPARQL Queries with LLMs

We use a recently published custom Bestiary KG [1] for our experiments, and use LLMs with knowledge cut-offs before the KG was made available and within our compute budget using an A16 GPU (16GB memory). These LLMs

[1] Source code: https://github.com/parklize/LLM4SPARQL2Q

Fig. 1. Example of generated question for a complex SPARQL query.



include `Llama2-7B`[2], `Llama3-8B`[3], `Mistral-7B`[4]. To evaluate how a fine-tuned LLM on one dataset (QALD-9-plus and QALD-10 [3,5]) performs on another dataset (Bestiary for our experiments), we also include a fine-tuned `Mistral-7B` (`Mistral-7B FT`) from recent work [4]. The Bestiary KG contains diverse information about more than 4k creatures from a fantasy role-playing game, comprising around 100k triples and 100 question-SPARQL query pairs.

We investigate n-shot prompting strategies with $n = 0, 1, 3, 5$. The prompt template is shown below, where `{n-shots}` serves as a placeholder for inserting n-shot examples, each consisting of a SPARQL query and its corresponding ground truth question. `{sparql}` indicates a placeholder for the target query.

```
######################### SYSTEM PROMPT ########################
You are SPARQL (SPARQL Protocol and RDF Query Language) expert.
Your goal is to analyze and interpret a given SPARQL query and
provide the corresponding question to lay users who do not have
any SPARQL knowledge and have no access to the query.

######################### USER PROMPT ##########################
Given a SPARQL query, provide the corresponding question at the
end. Your answer should be self-explanatory and should contain
only the corresponding question without any other additional text.

{n-shots}

The SPARQL query is between [SPARQL] and [/SPARQL] tags as below:
[SPARQL]
{sparql}
[/SPARQL]

The corresponding question to the SPARQL query:
```

[2] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[3] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[4] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

Table 1. BERT scores (Precision, Recall, and F1) using different LLMs and prompting strategies for generating the corresponding questions of SPARQL queries. The best scores using each LLM are underlined, and the overall best scores are in bold.

LLM	Prompting	Precision	Recall	F1
Llama2-7B	0-shot	0.9562	0.9673	0.9617
	1-shot	0.9612	0.9669	0.9640
	3-shot	0.9640	0.9672	0.9656
	5-shot	0.9649	0.9674	0.9661
	CoT	0.9591	0.9659	0.9624
Llama3-8B	0-shot	0.9598	0.9672	0.9635
	1-shot	0.9658	0.9672	0.9665
	3-shot	0.9651	0.9671	0.9661
	5-shot	0.9643	0.9664	0.9654
	CoT	0.9644	0.9670	0.9657
Mistral-7B	0-shot	0.9603	0.9667	0.9635
	1-shot	0.9654	0.9663	0.9658
	3-shot	0.9660	**0.9677**	0.9668
	5-shot	0.9657	0.9667	0.9662
	CoT	0.9618	0.9669	0.9643
Mistral-7B FT	0-shot	0.9669	0.9664	0.9666
	1-shot	0.9672	0.9661	0.9667
	3-shot	0.9670	0.9663	0.9666
	5-shot	**0.9676**	0.9666	**0.9671**
	CoT	0.9654	0.9665	0.9660

In addition, we also examine a 0-shot Chain-of-Thought (CoT) prompting strategy [6], which appends *"Take a deep breath and work on this problem step by step."* to the end of the prompt template to encourage LLMs think longer when deriving the corresponding question of a given query.

For evaluation, we opt for BERT scores [7], which measures the semantic similarity between generated and ground-truth questions rather than relying on word-level comparisons. As we observed in Fig. 1, the generated question may be phrased differently, yet it can still be semantically similar to the ground truth.

3 Results

Table 1 shows the BERT scores using different LLMs with n-shot prompting strategies ($n = 0, 1, 3, 5$) and `CoT`, with our discussion primarily focusing on the F1 score here. First, the results from the three pretrained LLMs show that using more shots ($n \geq 1$) leads to significant improvements compared to 0-shot for each LLM overall ($p < 0.05$). For `Llama3-8B`, using `CoT` outperforms `0-shot` significantly. In contrast, no significant differences are observed between `CoT` and `0-shot` for other LLMs. Overall, `Mistral-7B FT`, which is fine-tuned on a different dataset – QALD-9-plus and QALD-10 – performs best with an F1 score of 0.9671 with `5-shot`.

Table 2. Generated questions by `Mistral-7B` and `Mistral-7B FT` for the ground truth: "does Caypup have more special abilities than Aasimar?".

LLM	Generated question	F1
Mistral-7B	Which has more specific abilities, "caypup" or "aasimar"? (Answer will be 'True' if "caypup" has more specific abilities, and 'False' otherwise.)	0.9557
Mistral-7B FT	Which has more abilities: a Caypup or an Aasimar?	0.9735

Secondly, we also observe that although `Mistral-7B FT` is fine-tuned on a different dataset, it significantly improves the performance for `0-shot` and `CoT` without fine-tuning. Unlike pre-trained LLMs, the fine-tuned LLM with `0-shot` performs comparably to those using more shots or `CoT`. This suggests that fine-tuning on other datasets can help improve the performance on a new dataset for the same SPARQL2Q task. By comparing the questions generated with the fine-tuned model to the one without fine-tuning, we observe that the improvement stems from the similarity in generated question style between the fine-tuning dataset and the Bestiary dataset.

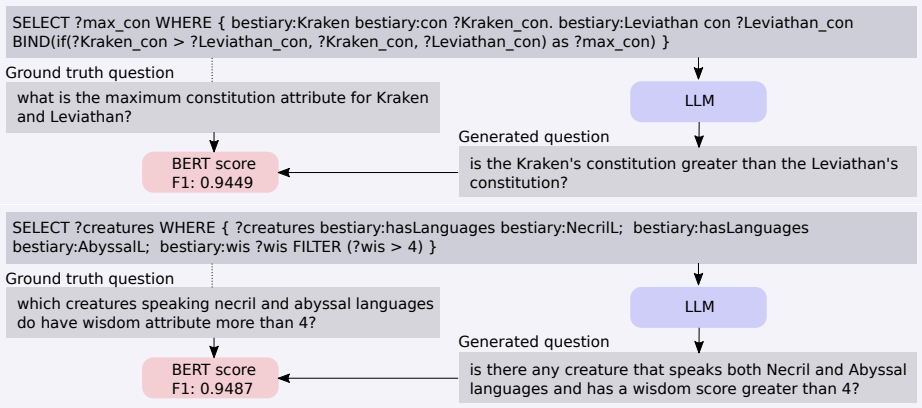
Table 2 illustrates an example where the ground truth question for a SPARQL query is: "does Caypup have more special abilities than Aasimar?". Since `Mistral-7B` in 0-shot setting is not familiar with the style of ground truth questions, its generated questions tend to be verbose as we can see from the table. In contrast, the fine-tuned model – `Mistral-7B FT` – generates a more concise, style-consistent question. This might also be one of the reasons why the performance of non-fine-tuned LLMs improves with more in-context examples ($n \geq 1$), as they begin to pick up on the desired response style.

To understand whether lower BERT scores reflect deviations in semantic meaning between generated and ground truth questions, Fig. 2 illustrates two examples with low BERT scores from the test set. The generated questions have meanings that differ significantly from the ground truth as well as the SPARQL queries, resulting in lower BERT scores. In contrast, as observed in Fig. 1, generated questions with different phrasing but similar semantic meaning can be reflected with higher BERT scores.

4 Conclusions and Future Work

The results indicate that, overall, more examples tend to help LLMs improve their understanding of the underlying questions of SPARQL queries. Zero-shot `CoT` does not provide significant improvement, and more research on how to design `CoT`, e.g., with well-crafted n-shots, can be explored in the future. Additionally, fine-tuned `Mistral-7B` on a different dataset also helps improve the performance on the Bestiary dataset of our interest. This is promising as it suggests that an open-source models can continue to evolve by being fine-tuned with additional datasets separately to enhance its performance on the same task.

Fig. 2. Examples with lower BERT scores reflecting lower semantic similarities.



Using BERT scores enables the use of a wide range of BERT variants. For future work, experimenting with other BERT variants for evaluation and incorporating human evaluations could provide a more robust measure of alignment between semantic similarity metrics and human judgment.

Acknowledgments. This work was funded by the GLACIATION (No. 101070141). The authors thank the ESWC reviewers for their constructive feedback, which helped improve both the presentation and interpretation of the experimental results.

References

1. Kovriguina, L., Teucher, R., Radyush, D., Mouromtsev, D.: Sparqlgen: One-shot prompt-based approach for sparql query generation. In: SEMANTiCS (2023)
2. Ngonga Ngomo, A.C., Bühmann, L., Unger, C., Lehmann, J., Gerber, D.: Sorry, i don't speak sparql: translating sparql queries into natural language (2013)
3. Perevalov, A., Diefenbach, D., Usbeck, R., Both, A.: Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers. In: ICSC. IEEE (2022)
4. Perevalov, A., Gashkov, A., Eltsova, M., Both, A.: Understanding sparql queries: Are we already there? multilingual natural language generation based on sparql queries and large language models. In: ISWC (2024)
5. Usbeck, R., Yan, X., Perevalov, A., Jiang, L., Schulz, J., Kraft, A., Möller, C., Huang, J., Reineke, J., Ngonga Ngomo, A.C., et al.: Qald-10–the 10th challenge on question answering over linked data: Shifting from dbpedia to wikidata as a kg for kgqa. Semantic Web **15**(6), 2193–2207 (2024)
6. Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q.V., Zhou, D., Chen, X.: Large language models as optimizers (2024), `https://arxiv.org/abs/2309.03409`
7. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019)