Text to SPARQL via LLMs fine-tuned with rewards and reasoning

Marcos Gôlo[1], Paulo Viviurka do Carmo[2], Edgard Marx[2], Ricardo Marcacini[1]

[1] Institute of Mathematical and Computer Sciences, University of São Paulo, São Carlos, São Paulo, Brazil
[2] Institute for Applied Informatics (InfAI) e. V., Leipzig, Saxony, Germany
marcosgolo@usp.br, carmo@infai.org, marx@infai.org, ricardo.marcacini@icmc.usp.br

Abstract. Large language models have made notable progress in translating natural-language questions into SPARQL queries for knowledge graphs. Recent studies from the first Text2SPARQL challenge have explored zero-shot and few-shot scenarios, as well as in-context learning techniques. With state-of-the-art (SoTA) results, the challenge exposes significant gaps, including limited use of open-source models and a scarcity of open-source fine-tuning, especially for reasoning. In this sense, we propose a new method for the Text2SPARQL task that combines pre-fine-tuning with explicit reasoning and a subsequent reward-based fine-tuning stage. First, we perform a fine-tuning to generate both a reasoning chain and the corresponding SPARQL query. Next, we apply reinforcement learning guided by reward functions. Our experiments show that the proposed model outperforms all baselines and six SoTA methods.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; **Machine learning algorithms**; *Reinforcement learning*; • **Information systems** → *Query languages*; Information retrieval.

Keywords: Text2SPARQL, Knowledge Graph Question Answering, Qwen3-Reasoning, Rewards Finetuning

1. INTRODUCTION

Knowledge graphs (KGs) have become an effective representation for organizing and linking structured information across diverse domains [Ji et al. 2021]. Large-scale resources such as DBpedia illustrate this potential by transforming Wikipedia content into interconnected knowledge that supports applications in artificial intelligence and information retrieval [Lehmann et al. 2015]. However, despite the large amount of available information, retrieving this knowledge remains challenging because querying DBpedia typically requires technical expertise in its ontology and SPARQL. This motivates the development of methods that translate natural-language questions into SPARQL queries, making KGs more accessible [Ali et al. 2022]. Figure 1 shows the Text2SPARQL task [Perevalov et al. 2024].

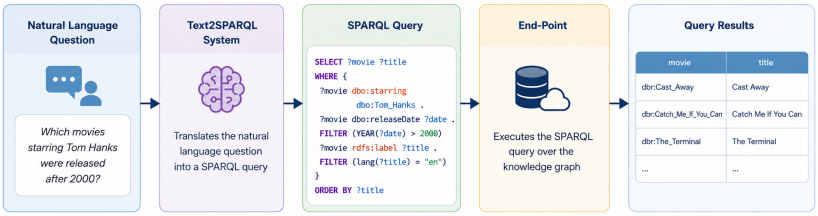

Fig. 1. Text2SPARQL task, in which a question is translated into a SPARQL query, executed over a KG.

Despite recent advances, Text2SPARQL remains challenging because models must generate queries that are both syntactically and semantically aligned with the specific KG, while also handling lexical ambiguity, entity and relation linking, KG-structural constraints, and multi-hop navigation [Soru et al. 2025]. Large Language Models (LLMs) have therefore become central to recent Text2SPARQL research, given their strong performance in natural language processing tasks. Recent works from the First International Text2SPARQL Challenge [Marx et al. 2025], have explored LLMs, considering

zero- and few-shot learning strategies [Wardenga and Kafer 2025; Brei et al. 2025; Dorsch et al. 2025], named entity recognition and retrieval-augmented generation [Berezin et al. 2025], experience pool generation and named entity linking [Perevalov and Andreas 2025], as well as fine-tuning [Soru et al. 2025]. However, some gaps remain, including limited use of open-source models, a scarcity of research on fine-tuning, and the lack of: (i) studies that incorporate reasoning and rewards during fine-tuning; and (ii) model comparisons on gold-standard benchmarks not seen during LLM pre-training.

We propose a novel reasoning LLM fine-tuned via a reward strategy to generate semantically correct SPARQL queries for the DBpedia KG. We enriched the training dataset with explicit reasoning annotations using a large proprietary model, enabling a two-stage fine-tuning procedure on a smaller open-source model. The first stage, a pre-fine-tuning phase, focuses on teaching the model to produce coherent reasoning to support query generation [Chang et al. 2023]. The second stage, guided by a reward strategy, aims to enhance the model's ability to generate syntactically valid and semantically accurate SPARQL queries [Guo et al. 2025]. We obtain a model capable of generating correct SPARQL queries and explaining how questions are transformed into executable queries, achieving SoTA results on the Text2SPARQL task over the DBpedia KG. Our contributions are:

(1) A novel open-source LLM with reasoning capabilities, fine-tuned with rewards (SoTA results);
(2) A novel enriched DBpedia dataset with reasoning attribute for reasoning finetuning;
(3) Experiments with open-source, proprietary LLMs, and SoTA methods in a benchmark.

2. RELATED WORK

Recent studies have increasingly explored LLM strategies to improve query generation over KGs. In the First International Text2SPARQL Challenge [Marx et al. 2025], several teams proposed methods that combine prompting, retrieval, entity linking, execution feedback, and agent-based reasoning. The **AIFB** team proposed a shape constraints-based approach that enriches LLM prompts with schema-level information from the KG [Wardenga and Kafer 2025]. Their pipeline first identifies candidate KG entities and then constructs few-shot prompts. The **INFAI** team proposed a method based on the SPINACH agent and the ReAct framework [Brei et al. 2025]. Their approach uses the LLM as a controller that can invoke search, inspection, and execution functions, including hybrid vector search with Qdrant and full-text search for retrieval. Similarly, the **IIS** team uses Llama and Qwen as controllers in two methods within a ReAct-based framework [Dorsch et al. 2025]. These methods rely on agent interaction, with actions such as keyword search, entity retrieval, and query execution.

The **MIPT** team introduced a multi-component pipeline that rewrites questions to reduce ambiguity, links entities to DBpedia, retrieves 2-hop neighborhoods to build local subgraphs, and retrieves similar question-query examples from external datasets [Berezin et al. 2025]. The **WSE** team proposed a two-stage plan-and-action workflow based on an offline experience pool and an online generation stage [Perevalov and Andreas 2025]. Their method uses named entity linking, in-context examples, and endpoint feedback to guide query generation. Finally, the **DBPEDIA** team proposed the only fine-tuning-based method in the challenge, training code-oriented LLMs such as CodeGen, StarCoder, and Code Llama on a refined subset of the NSpM dataset [Soru et al. 2025]. However, due to hardware limitations, only part of the planned training was completed, which affected convergence.

The approaches show strong interest in prompting engineering, in-context learning, retrieval, and execution-loop strategies. However, three main limitations remain. First, fine-tuning has received limited attention. Second, although several methods use execution feedback in loop-based pipelines, this feedback is not explicitly incorporated as a Reward signal during fine-tuning. Third, existing methods do not explicitly focus on teaching models to produce structured Reasoning for Text2SPARQL, even though reasoning-oriented LLMs have shown strong potential in sequential and compositional tasks such as code and query generation. To address these gaps, we present our proposal in the next section, which combines fine-tuning, explicit reasoning, and reward-based optimization.

3. TEXT2SPARQL THROUGH LLMS WITH REASONING AND REWARDS

We propose fine-tuning Large Language Models (LLMs) using reasoning and reward-based strategies [Liu et al. 2023; Chang et al. 2023; Shao et al. 2024; Guo et al. 2025]. We adopt a pre-fine-tuning phase focused on next-token prediction to improve text generation and reasoning [Liu et al. 2023; Chang et al. 2023], followed by a reward-based fine-tuning stage [Shao et al. 2024; Guo et al. 2025]. To enable this process, we enriched the dataset with a reasoning attribute and designed a novel reward strategy tailored for information retrieval in the DBpedia KG. We illustrate our proposal in Figure 2. Originally, we had a dataset comprising natural-language questions and their corresponding SPARQL queries. After, we enrich the dataset by adding a reasoning attribute, yielding a final enriched dataset ready for reasoning fine-tuning.

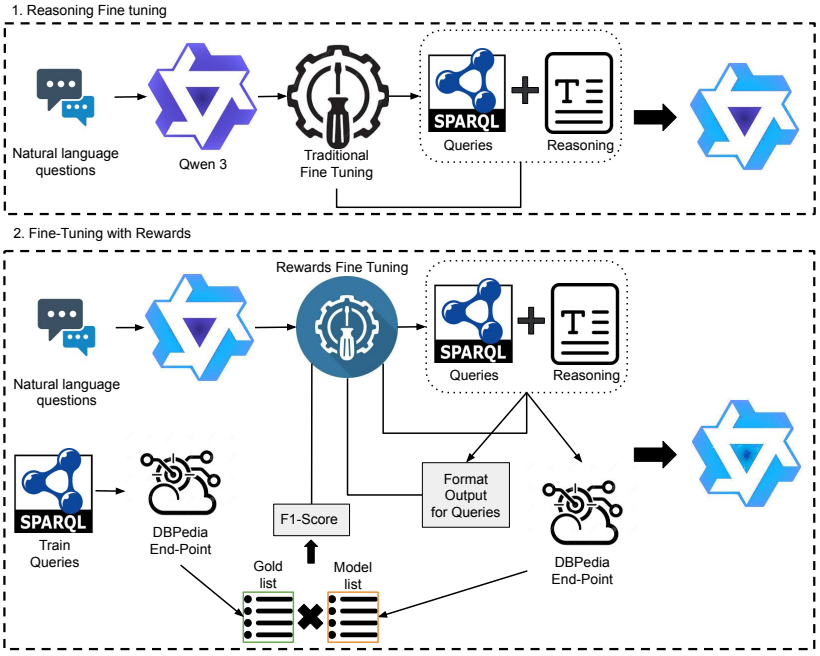

Fig. 2. Our proposal for fine-tune LLMs with reasoning and rewards to perform information retrieval.

The proposal begins with a pre-fine-tuning process to enhance reasoning capabilities by generating a SPARQL query for the LLM, yielding two different outputs [Shao et al. 2024; Guo et al. 2025]. The system prompt incorporates special tokens that explicitly divide the model output into these two parts: the reasoning component, which provides a textual explanation of the transformation process, and the solution component, which contains the executable query. Thus, at this stage, the input is a natural-language question, and the output is structured text containing both the reasoning and the corresponding SPARQL query, each represented as separate token strings [Liu et al. 2023]. We explore the traditional supervised fine-tuning, where the model is trained to predict the following tokens of the output sequence o, given an input text q [Chang et al. 2023]. The objective is to maximize the probability of generating the correct tokens conditioned on the preceding ones. The loss function used in this pre-fine-tuning stage is defined as [Chang et al. 2023]: $\mathcal{L}(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \log \pi_\theta(o_t \mid q, o_{<t})$, where T is the number of tokens generated, q is the input prompt, and π_θ is the LLM model that estimates the probability distribution for the next token.

Meanwhile, in the reward-based fine-tuning stage, our objective is to refine the model by rewarding the correct information retrieval from the DBpedia KG when comparing to the original query. This stage builds upon the model previously trained in the pre-fine-tuning phase. We propose using the

Group Relative Policy Optimization (GRPO) reinforcement learning algorithm [Shao et al. 2024; Guo et al. 2025] for this fine-tuning phase. GRPO is a reinforcement learning approach that iteratively improves the model by leveraging data generated during training. Its primary goal is to give input and maximize the relative advantage of the responses produced in the final task, information retrieval through SPARQL. The GRPO process can be divided into three main stages: response generation, advantage calculation, and loss computation. At each training iteration, we generate G candidate responses for each prompt. In our case, each response consists of structured text output that includes a reasoning component and a corresponding SPARQL query, both of which are related to the given question. We denote each generated output as o_i [Shao et al. 2024].

For each of the G generated responses, we compute the output from two complementary reward functions: an accuracy reward (ar_i) and a format reward (fr_i). ar_i evaluates whether the generated SPARQL query retrieves the correct answer from the KG, while the format reward ensures that the output is structured in the expected evaluation format. In particular, the fr_i encourages the model to produce reasoning enclosed between two special tokens and the SPARQL query enclosed between another pair of special tokens. Our proposed accuracy reward assesses whether the generated SPARQL query returns the same list of answers as the true query, i.e., a correct information retrieval task from the KG. We first extract the SPARQL portion from the generated output o_i, separating it from the reasoning component. The query is then executed against the official DBpedia SPARQL endpoint used for evaluation. The result list from the generated query is compared with the list produced by the ground truth query. Using these two sets of answers, we compute the f_1-score, which serves as the core of our accuracy reward function. This function follows three rules: (1) if the query returns an empty result or is syntactically invalid, we assign a reward of 0; (2) if the query is valid but the f_1-score is 0, a penalty of -4.5 is applied; and (3) if the f_1-score is greater than 0, we return $5 \cdot f_1\text{-score}(o_i)$:

$$fr_i = \begin{cases} 3 & \text{if } o_i \text{ is formatted,} \\ 0 & \text{otherwise.} \end{cases} \quad (1) \qquad ar_i = \begin{cases} 0 & \text{if empty list,} \\ -4.5 & \text{if } f_1\text{-score}(o_i) = 0 \\ 5 \cdot f_1\text{-score}(o_i) & \text{otherwise.} \end{cases} \quad (2)$$

The model is trained by generating G responses for each question, with $G > 1$. The advantage is computed to enable relative comparisons among the G responses, thereby aligning the reward-based fine-tuning process with advantage estimation [Guo et al. 2025]. The normalized advantage calculation is presented in Equations 3 and 4. The advantage is obtained as the sum of the two rewards ($A_i = \text{Ar}_i + \text{Fr}_i$). We used the default loss values from the *unsloth* tutorial for fr_i and ar_i.

$$Ar_i = \frac{ar_i - \text{mean}(ar_1, ar_2, \cdots, ar_G)}{\text{std}(ar_1, ar_2, \cdots, ar_G)} \quad (3) \qquad Fr_i = \frac{fr_i - \text{mean}(fr_1, fr_2, \cdots, fr_G)}{\text{std}(fr_1, fr_2, \cdots, fr_G)} \quad (4)$$

After defining the reward functions, the objective of fine-tuning with GRPO is to maximize the advantage A_i, while ensuring that the model remains close to the reference policy through the use of the Kullback–Leibler (KL) divergence between the reference model (trained during the pre-fine-tuning stage) and the current model [Shao et al. 2024]. Consequently, the loss function is defined as follows:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G} \sum_{i=1}^{G} \Big[\min \Big(\Delta_i \, A_i, \text{clip} \Big(\Delta_i, 1-\epsilon, 1+\epsilon \Big) A_i \Big) - \beta \, \text{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}) \Big], \quad (5)$$

where $\Delta_i = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}$, $\text{clip}\,(\cdot, 1-\epsilon, 1+\epsilon)$ ensures that updates do not deviate excessively from the reference policy by constraining the policy ratio. Specifically, q denotes the input prompt—comprising the system prompt and the question in our case; π_θ represents the current trained model; $\pi_{\theta_{\text{old}}}$ refers to the model from the previous iteration; π_{ref} corresponds to the model trained during the pre-fine-tuning; β and ϵ are hyperparameters; and $\mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$ is [Shao et al. 2024]:

$$\mathbb{D}_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}) = \frac{\pi_{\mathrm{ref}}(o_i \mid q)}{\pi_\theta(o_i \mid q)} - \log \frac{\pi_{\mathrm{ref}}(o_i \mid q)}{\pi_\theta(o_i \mid q)} - 1 \tag{6}$$

4. EXPERIMENTAL EVALUATION

This section presents the experimental evaluation of this article. We present the dataset used to train the LLM, the test dataset (gold-standard Text2SPARQL), the experimental settings, the results, and the discussion. Our goal is to show that our reward-based reasoning fine-tuning approach outperforms existing SoTA methods. The experimental evaluation codes are publicly available[1].

4.1 Datasets

In recent years, several datasets have been proposed for converting text into SPARQL queries, with DBpedia as the underlying knowledge base. Among the datasets used in previous studies, and those available on Hugging Face, we highlight four primary sources that were used to construct our training set for LLM fine-tuning: QALD1–9, LC-QuAD 1.0, ParaQA-SPARQL-to-Text, and Question-SPARQL[2]. For the training dataset, we merged the four datasets into a single dataset. Although some of the datasets include questions in multiple languages, we retained only English and Spanish queries, as they were the primary focus of this study. A preprocessing pipeline was applied to ensure that the training data reliably reflected the constraints and characteristics of the target knowledge base. This pipeline consisted of three main steps. First, we filtered out inconsistent SPARQL queries that were not syntactically valid, i.e., failed to execute on the DBpedia endpoint used for evaluation. Second, we removed duplicate queries, i.e., question-query pairs that appeared in more than one of the four datasets. After executing these two steps, we achieved a training set with 13,000 question-query pairs. For the test, we use the benchmark from the Text2SPARQL challenge. The challenge introduced 200 new bilingual (English and Spanish) question-query pairs targeting the DBpedia core version from October 2015. The set includes 200 questions to 100 unique SPARQL queries, each paired with human-curated English and Spanish versions [Marx et al. 2025].

4.2 Experimental Settings

We evaluate several open-source LLMs as zero-shot and fine-tuned baselines, including LLaMA 3.1 8B, LLaMA 3.3 70B, Phi-4 14B, and Qwen 2.5 models with 7B, 14B, and 32B parameters. Fine-tuning is performed with Unsloth using QLoRA, for 100 steps with a learning rate of 0.001. To generate reasoning annotations, we compare GPT-4o-mini, GPT-4o, GPT-4.1-mini, and GPT-4.1 in a zero-shot setting and select the best-performing model to produce step-by-step reasoning traces for the training set. These proprietary models also serve as strong baselines. During the inference phase, we evaluated model performance at four temperatures (0.01, 0.25, 0.5, and 0.75) to analyze the impact of temperature variability on SPARQL query generation. Futhermore, we compare our proposal with six methods from first text2SPARQL challenge[Marx et al. 2025]: **INFAI** [Brei et al. 2025], **MIPT** [Berezin et al. 2025], **IIS-L** and **IIS-Q** [Dorsch et al. 2025], **AIFB** [Wardenga and Kafer 2025], and **WSE** [Perevalov and Andreas 2025]. For our finetuning proposal, we use Qwen 3 (4B and 14B) [Yang et al. 2025]. We do not include the results from the **DBPEDIA** team [Soru et al. 2025] because the proposed methods achieved $f_1 = 0$. All generated queries are executed on the local DBpedia endpoint provided by the challenge, and the returned answer lists are evaluated with f_1-score using Pytrec_eval. We ran our experiments on an NVIDIA RTX L40S.

[1] `https://github.com/GoloMarcos/Text2SPARQL-Reasoning`
[2] `https://huggingface.co/datasets/aksw/Text2SPARQL-Raw`

4.3 Results and Discussion

We organize our results into three stages. The first stage consists of fine-tuning LLMs without reasoning. This step aims to measure the ability of open-source LLMs to generate SPARQL queries through Fine-tuning and to establish baseline results. The second stage involves evaluating larger proprietary LLMs against the open-source baselines to select the model to enrich the training dataset with reasoning for the proposed method. The final stage compares the proposed model with the proprietary LLM used to generate the reasoning, the best-performing baseline, and the approaches from the Text2SPARQL Challenge. All f_1 values reported refer to the gold standard test set used in the first international Text2SPARQL Challenge. Table I presents the results from the first stage, where we compare three LLM families, LLaMA, Qwen, and Phi, with varying parameter sizes for SPARQL query generation, with inference performed at four temperatures. Each row reports the performance of a model across the scenarios represented by the columns. The best result is highlighted in bold, and the second-best is underlined. We also compared the results of our baseline models across different temperatures to determine whether temperature positively influences SPARQL generation.

Table I. f_1 results for 0-shot and fine-tuned (FT) models with each temperature. Best values are in bold for each model.

Models	Zero-Shot	FT (0.01)	FT (0.25)	FT (0.50)	FT (0.75)
Llama (8b)	0.027	**0.319**	0.311	0.308	0.302
Llama (70b)	0.134	**0.409**	0.376	0.358	0.387
Qwen (7b)	0.129	0.310	**0.320**	0.291	0.269
Qwen (14b)	0.264	0.374	0.381	**0.410**	0.349
Qwen (32b)	0.152	0.395	**0.409**	0.407	0.370
Phi (14b)	0.221	0.336	0.336	0.336	0.336

The Phi model, regardless of temperature, produces the same result. Thus, in scenarios where reproducibility is essential, using Phi is recommended. We observed that a high temperature of 0.75 hinders the model's ability to generate SPARQL queries. On the other hand, other temperature values can be beneficial, but their impact depends on the LLM family. For the Llama models, lower temperatures, such as 0.01, yielded the best results. For the Qwen models, medium temperatures such as 0.25 and 0.5 outperformed the others. Therefore, when using these models, we recommend selecting temperature values based on each model's optimal performance, rather than applying a single value across all models. The pre-trained LLaMA (8 billion-parameter) model achieved the worst result. On the other hand, the best performance was achieved by the Qwen model, which had 14 billion parameters. The second-best result was a tie between the LLaMA model with 70 billion parameters and the Qwen model with 32 billion. Considering the open-source LLMs explored, the best results were obtained from fine-tuning, followed by the pre-trained models, which yielded the lowest f_1.

We explored four pre-trained models from the GPT family to assess how our open-source baselines perform relative to proprietary models. GPT-4o-mini obtains 0.35 of f_1, GPT-4o obtains 0.323, GPT-4.1-mini obtains 0.461, and GPT-4.1 obtains 0.487. We observed that GPT-4o-mini and GPT-4o underperformed relative to our best fine-tuned baseline (Qwen, with $f_1 = 0.41$). On the other hand, GPT-4.1-mini and GPT-4.1 achieved better results, with GPT-4.1 in particular reaching an f_1-score of 0.487. Therefore, because GPT-4.1 outperformed our simple open-source fine-tuning, and given the advantages of leveraging pre-training knowledge from a very large language model, we chose it to generate the reasoning attribute.

Figure 3 presents the results of the third stage, where we compare our proposed reasoning-rewards approach against the models from the challenge, our weak baseline (open-source), and our strong baseline (proprietary). Our proposal is based on the RR models (Reasoning-Rewards), where RR^1 uses the Qwen 3 model with 14 billion parameters as the base, and RR^2 uses the Qwen 3 model with 4 billion parameters as the base. We organized our results into four groups: blue bars represent our

RR models, orange bars represent our weak baseline, purple bars represent our strong baseline, and green bars represent the results of the Text2SPARQL challengers' methods. Our RR[1] model achieved the best overall performance, while the WSE team's method obtained the second-best result. The lowest scores were observed with the IIS team's approach. Notably, our weak baseline outperformed most challenge participants, ranking third behind the WSE and AIFB teams. Meanwhile, our strong baseline, the pretrained GPT-4.1, ranked second among the challenge methods, only behind WSE, securing the third-best overall result. Furthermore, our smaller-parameter model, RR[2], ranked second among the challenge methods, delivering competitive results against our strong baseline and ultimately achieving the fourth-best overall result.

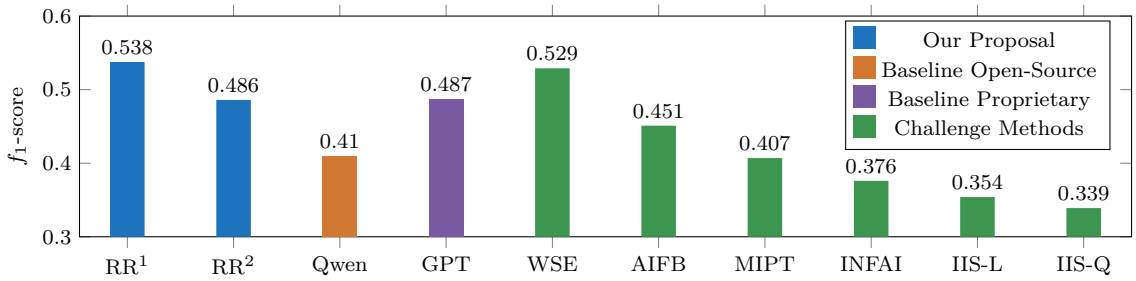

Fig. 3. Comparison of SoTA models with our reasoning model. We present the f_1-score for the methods.

We observed that using a larger model for our reasoning-rewards approach improved performance by 5%, indicating that if the priority is accuracy, the 14B model is the preferred choice. However, the 4B model still achieves competitive results, making it a viable alternative when smaller models are required. Another noteworthy point is GPT-4.1's promising performance in a simple zero-shot setting, highlighting the power of proprietary models. Nevertheless, their use requires per-token payment, which may be impractical in some scenarios. With reasonably available hardware, our 4-billion-parameter model can reproduce GPT-4.1's results, demonstrating the effectiveness of incorporating both the reasoning stage and reward-based fine-tuning in a smaller model. The most competitive baseline was achieved by the WSE team. Their pipeline focused on two main stages that closely align with our strategy, but relied on proprietary models. Their approach employed a planning or in-context learning step (similar to reasoning), in which GPT-4o was used to design the SPARQL generation. They also incorporated an Experience Pool stage to generate training queries and verify that the returned list was correct. The key difference is that we trained a model directly with these strategies, while the authors applied them in an offline stage of their method to build an experience pool that guided the model during the online phase.

Contextualizing the model through reasoning and employing reward strategies or tracking successes and errors in an experience pool yield strong results in SPARQL generation with LLMs, especially when the model is trained with these strategies. Considering the WSE team's competitive performance, we emphasize that each method and strategy has its own advantages and disadvantages. The WSE team's approach does not have strong hardware requirements, since it leverages proprietary models executed elsewhere. However, their usage cost during the challenge was around 3 USD per 100 executed queries. In addition to ongoing costs, accessing proprietary models executed on their hardware can compromise sensitive data and might not be possible in all scenarios. In contrast, our approach was developed for local execution, offering full accessibility while remaining open, free, and privacy-preserving. We can observe the negative impact of the lack of evaluation of individual pipeline stages in the challenge methods (e.g., through an ablation study). Notably, even with multiple steps applied before and during SPARQL query generation, all methods, except for the WSE results, underperformed compared to using GPT-4.1-mini in a zero-shot setting, i.e., the simplest version of an LLM with a prompt requesting the SPARQL query for a given question.

5. CONCLUSION AND FUTURE WORK

We investigated the Text2SPARQL task on the DBpedia KG, exploring a new gold-standard benchmark and an innovative method that combines pre-fine-tuning with explicit reasoning, followed by a reward-based stage. Our approach bridges gaps in the literature, including the limited use of open-source models and the scarcity of strategies that incorporate explicit reasoning and reward signals during fine-tuning. Our experiments demonstrated that the proposed method outperforms baselines, proprietary models, and six SoTA approaches. We found that incorporating explicit reasoning, combined with reinforcement learning via GRPO, significantly improves f_1. As future work, we intend to conduct an ablation study to separate the contributions of the reasoning component and reward-based fine-tuning. Also, we intend to investigate the integration of retrieval-augmented generation with similarity search to reduce entity errors. We also aim to compare our method with methods from the second edition of the Text2SPARQL challenge. Finally, we intend to explore more advanced reinforcement learning strategies.

Acknowledgments

This study was partially funded by FAPEC (23104.003594/2025-12), CNPq (317002/2026-0), FAPESP (2025/07160-0, 2024/08485-8, 2024/15430-5), and SMARTY Project (CHIPS Joint Undertaken as part of the European Union's Horizon Europe research and innovation programme, 101140087).

REFERENCES

Ali, W., Saleem, M., Yao, B., Hogan, A., and Ngomo, A.-C. N. A survey of rdf stores & sparql engines for querying knowledge graphs. *The VLDB Journal*, 2022.

Berezin, D., Avdeev, R., and Somov, O. Airi team in text2sparql challenge: Text-to-sparql executor for question-answering over knowledge graphs. In *First International TEXT2SPARQL Challenge*. CEUR, 2025.

Brei, F., Bʹuhmann, L., Frey, J., Gerber, D., Meyer, L.-P., Stadler, C., and Bulert, K. Aruqula - an llm based text2sparql approach using react and knowledge graph exploration utilities. In *First International TEXT2SPARQL Challenge*. CEUR, 2025.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.

Dorsch, R., Henselmann, D., and Harth, A. Graf von data: A knowledge graph question answering agent for organisational usage. In *First International TEXT2SPARQL Challenge*. CEUR, 2025.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems* 33 (2): 494–514, 2021.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web* 6 (2): 167–195, 2015.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. Gpt understands, too. *AI Open*, 2023.

Marx, E., do Carmo, P., Gôlo, M., and Trump, S., editors. *Preface of the First International TEXT2SPARQL Challenge (TEXT2SPARQL'25)*. CEUR, 2025.

Perevalov, A. and Andreas, B. Text-to-sparql goes beyond english: Multilingual question answering over knowledge graphs through human-inspired reasoning. In *First International TEXT2SPARQL Challenge*. CEUR, 2025.

Perevalov, A., Both, A., and Ngonga Ngomo, A.-C. Multilingual question answering systems for knowledge graphs–a survey. *Semantic Web* 15 (5): 2089–2124, 2024.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Soru, T., Joshi, S., Tiwari, S., Shahinmoghadam, M., and Panchbhai, A. Question answering over dbpedia with fine-tuned autoregressive models. In *First International TEXT2SPARQL Challenge*. CEUR, 2025.

Wardenga, J. G. and Kafer, T. Leveraging data shapes in large language model contexts for question answering on public and private knowledge graphs. In *First International TEXT2SPARQL Challenge*. CEUR, 2025.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.