# The Role of Exploration Modules in Small Language Models for Knowledge Graph Question Answering

**Yi-Jie Cheng**[1,2] **Oscar Chew**[1] **Yun-Nung Chen**[2]
[1]ASUS
[2]National Taiwan University
b09202004@ntu.edu.tw oscar_chew@asus.com y.v.chen@ieee.org

## Abstract

Integrating knowledge graphs (KGs) into the reasoning processes of large language models (LLMs) has emerged as a promising approach to mitigate hallucination. However, existing work in this area often relies on proprietary or extremely large models, limiting accessibility and scalability. In this study, we investigate the capabilities of existing integration methods for small language models (SLMs) in KG-based question answering and observe that their performance is often constrained by their limited ability to traverse and reason over knowledge graphs. To address this limitation, we propose leveraging simple and efficient exploration modules to handle knowledge graph traversal in place of the language model itself. Experiment results demonstrate that these lightweight modules effectively improve the performance of small language models on knowledge graph question answering tasks. Source code: https://github.com/yijie-cheng/SLM-ToG/.

## 1 Introduction

Large Language Models such as GPT4 (OpenAI, 2024), Gemini (Google, 2024), Qwen (Bai et al., 2023) have achieved state-of-the-art performance across a wide range of natural language processing tasks. Despite their impressive capabilities, a key limitation is the lack of interpretability in their decision-making processes. Moreover, they are prone to hallucination, especially when the required knowledge is not present in their parametric memory. To tackle these challenges, Think-on-Graph (ToG; Sun et al., 2024) treats the LLM as an agent that dynamically interacts with knowledge graphs to retrieve external knowledge, exemplifying a LLM×KG paradigm that has garnered significant attention. To cast LLMs as an agent, ToG and similar approaches typically rely on very large models (Xu et al., 2024; Cheng et al., 2024; Liang and Gu, 2025), limiting their accessibility for low-resource settings. Other recent efforts (Luo et al., 2024; He et al., 2024; Ao et al., 2025; Yang et al., 2025) have proposed additional reasoning or exploration modules to improve LLM-KG integration, but these methods require task-specific training or fine-tuning.

In this paper, we focus on a practical setting where end users or system deployers have access only to small- or medium-sized language models for inference. In this context, an important question arises: how effectively can these SLMs leverage knowledge graphs for question answering? To explore this, we examine Think-on-Graph (Sun et al., 2024), a representative training-free framework, and observe that when applied to SLMs rather than LLMs, ToG underperforms and sometimes even falls behind the Chain-of-Thought (CoT) baseline (Wei et al., 2022). Through detailed analysis, we attribute this failure to the SLMs' limited ability to explore and reason over knowledge graphs. We argue that using lightweight passage retrieval methods such as SentenceBERT and GTR for exploration can substantially enhance the effectiveness of knowledge graph traversal for SLMs. We would like to point out that the novelty of our work does not lie in introducing new models or architectures. Rather, we revisit previously underestimated techniques and demonstrate their effectiveness in enhancing reasoning performance in resource-constrained settings. Our contributions can be summarized as follows:

- We demonstrate that the existing ToG framework is not as effective for SLMs in KGQA.
- We identify the exploration stage as a key bottleneck for SLM performance in knowledge graph reasoning.
- We show that incorporating simple and efficient passage retrieval modules significantly improves SLMs' ability to traverse and reason over knowledge graphs.

## 2 Traversing Knowledge Graphs with Small Language Models

### 2.1 Preliminaries

Think-on-Graph (Sun et al., 2024) is a framework for KGQA that casts a language model as an agent navigating a knowledge graph to perform multi-hop reasoning. It operates in three main stages:

- Initialization: The model extracts topic entities from the input question and locates them in the KG to form initial reasoning paths.
- Exploration: Using beam search, the model iteratively expands these paths by exploring neighboring relations and entities. At each step, the LLM ranks candidates and prunes less relevant options, guided by the question context.
- Reasoning: Once sufficient evidence is gathered, the LLM generates a final answer based on the maintained reasoning paths.

This structured interaction enables interpretable and context-sensitive reasoning while leveraging the strengths of both KGs and language models.

### 2.2 Exploration Modules for SLMs

In Section 3.3, we will show that SLMs are less effective for KGQA due to their limitation in exploration stage. To address the weaknesses of using only SLM itself for exploration of KG, we examine the use of simple, efficient retrieval models in Section 3.4. These models, which measure semantic similarity between text segments, have shown strong performance in passage retrieval tasks and hence are well-suited to assist SLMs in pruning irrelevant candidates during KG traversal. Importantly, they can be used in a zero-shot, plug-and-play manner, requiring no additional training or fine-tuning, making them well-suited for low-resource settings.

**Classic Retrieval Index** BM25 (Robertson and Zaragoza, 2009) is a ranking function used in information retrieval that scores how well a document matches a query based on term frequency and how common the term is across all documents.

**Dense Retrieval** We consider two dense retrievers: SentenceBERT (Reimers and Gurevych, 2019), a BERT-based model fine-tuned for producing semantically meaningful sentence embeddings, and GTR (Ni et al., 2022), a T5-based model optimized for passage retrieval tasks. Both models have approximately 110 million parameters which is substantially smaller than the smallest SLM (0.5B) evaluated in this work. Implementation details are presented in Appendix. A.

| Models | | CWQ | WebQSP |
|---|---|---|---|
| *Large Language Models* | | | |
| GPT-4.1 | w/ CoT | 0.505 | 0.765 |
| | w/ ToG | **0.575** | **0.810** |
| *Small Language Models* | | | |
| Qwen2-0.5b | w/ CoT | 0.170 | **0.345** |
| | w/ ToG | **0.175** | 0.210 |
| Gemma2-2b | w/ CoT | 0.185 | **0.465** |
| | w/ ToG | **0.255** | 0.420 |
| Phi-3-mini-3.8b | w/ CoT | **0.385** | **0.530** |
| | w/ ToG | **0.385** | 0.515 |
| Qwen2-7b | w/ CoT | 0.355 | 0.555 |
| | w/ ToG | **0.395** | **0.630** |
| Llama-3-8b | w/ CoT | 0.385 | **0.660** |
| | w/ ToG | **0.395** | 0.620 |
| Mean SLM | w/ CoT | 0.296 | **0.511** |
| | w/ ToG | **0.321** | 0.479 |

Table 1: Comparison of ToG and CoT across model sizes. While ToG substantially improves GPT-4.1, its effectiveness does not consistently extend to SLMs.

## 3 Experiments

In this section, we aim to answer the following research questions:

- **RQ1**: How do SLMs perform in KGQA compared to a larger proprietary LLM (GPT-4.1)?
- **RQ2**: Why are SLMs less effective at leveraging KGs for question answering tasks?
- **RQ3**: How effective are SLMs when paired with better-suited exploration modules?

### 3.1 Setup

**Datasets and Metrics** Following Sun et al. (2024), we use Freebase (Bollacker et al., 2008) as our underlying knowledge graph. We evaluate our models on two benchmark datasets: ComplexWebQuestions (CWQ; Talmor and Berant, 2018) and WebQSP (Yih et al., 2016). CWQ contains complex questions that require up to 4-hop reasoning while WebQSP which primarily involves 1- to 2-hop reasoning tasks. To reduce computational cost,

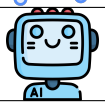
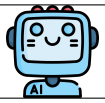
| **Question:** What type of government is used in the country with Northern District? | |
|---|---|
| **With knowledge triplets retrieved by SLM** | |
| | ('Northern District', 'country', 'Israel'),<br>('Northern District', 'administrative_parent', 'Israel') |
|  | **SLM:** The triplets do not provide information about the type of government used in Israel. |
| **With knowledge triplets retrieved by GPT4.1** | |
| | ('Northern District', 'country', 'Israel'),<br>('Northern District', 'administrative_parent', 'Israel'),<br>('Israel', 'form_of_government', 'Parliamentary system'),<br>('Israel', 'administrative_children', 'Northern District') |
|  | **SLM:** Based on the given knowledge triplets, the country with the Northern District is Israel, which uses a Parliamentary system as its form of government. |

Table 2: An example illustrating the limitations of an SLM when performing KG exploration on its own. When relying solely on its retrieved triplets, the SLM fails to answer the question. However, when provided with triplets retrieved by GPT-4.1, including the key relation, the same SLM is able to produce the correct answer.

| Models | CWQ | WebQSP |
|---|---|---|
| Qwen2-0.5b CoT | 0.170 | 0.345 |
| w/ GPT-4.1 ToG | **0.430** | **0.610** |
| Gemma2-2b CoT | 0.185 | 0.465 |
| w/ GPT-4.1 ToG | **0.430** | **0.690** |
| Phi-3-mini-3.8b CoT | 0.385 | 0.530 |
| w/ GPT-4.1 ToG | **0.520** | **0.745** |
| Qwen2-7b CoT | 0.355 | 0.555 |
| w/ GPT-4.1 ToG | **0.520** | **0.765** |
| Llama-3-8b CoT | 0.385 | 0.660 |
| w/ GPT-4.1 ToG | **0.550** | **0.805** |
| Improvement w/ GPT4.1 | 0.970 | 1.060 |

Table 3: Performance of SLMs with GPT-4.1-assisted exploration. With high-quality context, SLMs can offer better improvement over the CoT baseline, highlighting exploration as the key bottleneck in the ToG framework

we sample 200 questions from each dataset for evaluation. We use exact match (EM) score as the primary evaluation metric, which measures whether the predicted answer string exactly matches the given answer.

**Language Models** We consider SLMs ranging in size from 0.5B to 8B parameters. The models include Qwen2 0.5B (Yang et al., 2024), Gemma2-2b (Team et al., 2024), Phi-3-Mini-3.8B (Abdin et al., 2024), Qwen2 7b and LLaMA 3-8B (Grattafiori et al., 2024).

### 3.2 RQ1: Think-on-Graph with LLMs and SLMs

We begin by examining the effectiveness of applying ToG to SLMs in comparison to LLMs. As shown in Table 1, while a giant LLM (GPT-4.1) [1] enjoys significant boost from ToG, we observe that SLMs equipped with ToG receive limited improvement and can perform even worse than the CoT baseline. This discrepancy underscores a key limitation: while ToG is effective for LLMs, its effectiveness does not translate well to the lower-capacity SLMs with weaker reasoning capabilities.

### 3.3 RQ2: Bottleneck of Exploration

Given that ToG fails to improve performance for SLMs, we further investigate the underlying cause. Our hypothesis is that, without effective exploration, SLMs lack access to the necessary information required to generate correct answers, resulting in low EM scores. To verify this, we test an upper bound where we temporarily assume the access to GPT-4.1 for exploration only. That is, GPT-4.1 is used to explore the knowledge graph and provide context to the SLMs to reason the final outputs. We first look into failure cases of SLMs and found that SLMs could not generate the correct answer due to lack of proper context, as illustrated in Table 2 [2]. As shown in Table 3, with the context provided by GPT-4.1, SLMs are able to reason effectively

[1] We use the GPT-4.1 snapshot released on April 14, 2025.
[2] The figure contains resources from Flaticon.com

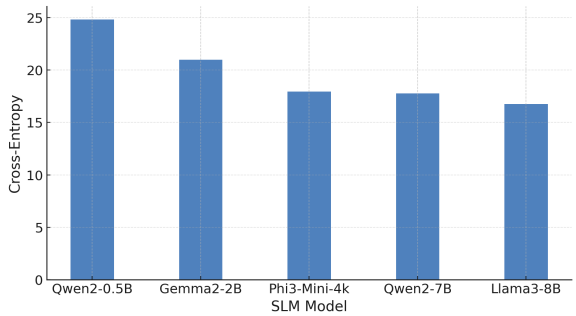



Figure 1: Cross-entropy alignment between the exploration outputs of SLMs and GPT-4.1 across different model sizes. A lower cross-entropy value indicates a closer alignment with GPT-4.1's exploration decisions. The consistent improvement with increasing model size highlights the critical role of exploration quality as the performance bottleneck for SLMs in the ToG framework.

and offer better improvement over the original CoT baseline.

We further treat the exploration outputs of GPT-4.1 as pseudo-ground truth and measure how closely the outputs of SLMs align with them in terms of cross-entropy. As shown in Figure 1, this alignment increases consistently with model size, supporting the view that exploration quality is a key bottleneck for SLMs within the ToG framework.

One might ask whether the difference in performance between SLMs and LLMs are due to their abilities in adhering to the questions/answer format. We have ruled out this possibility by leveraging Constrained Decoding. Relevant details are presented in Appendix B.

### 3.4 RQ3: Passage Retrieval for Exploration

As we have determined in Section 3.3 the core limitation of SLMs in the ToG framework lies in their inadequate performance during the exploration stage. One promising direction to address this is to decouple the exploration process from the language model itself. Instead of relying on the SLM to retrieve relevant knowledge paths, we explore the use of lightweight passage retrieval models to assist in this stage. These models are efficient, require no additional training, and have shown strong performance in passage retrieval tasks, making them a natural fit for supporting KG exploration. We present our main results in Table 4. Across all SLMs we studied, SentenceBERT and GTR obtain substantial improvement over both the original ToG and CoT for SLMs. This result highlights the effectiveness of leveraging passage retrieval models to assist SLMs during exploration. Interestingly, our findings contrast with those of Sun et al. (2024), who report that integrating passage retrieval models leads to significant performance degradation when applied to LLMs instead of SLMs. We further discuss this in Appendix C.

| Models | CWQ | WebQSP |
|---|---|---|
| Qwen2-0.5b ToG | 0.175 | 0.210 |
| w/ BM25 | 0.130 | 0.285 |
| w/ SentenceBERT | **0.210** | **0.295** |
| w/ GTR | 0.120 | 0.250 |
| Gemma2-2b ToG | 0.255 | 0.420 |
| w/ BM25 | 0.205 | 0.425 |
| w/ SentenceBERT | 0.250 | **0.590** |
| w/ GTR | **0.275** | 0.570 |
| Phi-3-mini-3.8b ToG | 0.225 | 0.515 |
| w/ BM25 | 0.370 | 0.500 |
| w/ SentenceBERT | **0.400** | 0.590 |
| w/ GTR | **0.400** | **0.620** |
| Qwen2-7b ToG | 0.395 | 0.630 |
| w/ BM25 | 0.360 | 0.550 |
| w/ SentenceBERT | 0.410 | **0.680** |
| w/ GTR | **0.430** | 0.675 |
| Llama-3-8b ToG | 0.395 | 0.620 |
| w/ BM25 | 0.390 | 0.500 |
| w/ SentenceBERT | **0.445** | 0.690 |
| w/ GTR | 0.400 | **0.700** |

Table 4: Effectiveness of lightweight passage retrieval methods for KG Exploration. SentenceBERT and GTR provides strong performance gains across models, validating its effectiveness for SLM-based KGQA.

## 4 Conclusion

In this paper, we investigate the limitations of SLMs in leveraging knowledge graphs for question answering. We identify the core issue as the inadequacy of SLMs in the exploration stage, where they often fail to retrieve accurate reasoning paths and relevant knowledge. To address this, we propose replacing the exploration component in ToG with lightweight passage retrieval models. Experiment results demonstrate that this approach not only improves the efficiency of the reasoning process but also enables SLMs to benefit more effectively from KGs. These findings may serve as a foundation for future research on more effective and accessible use of KGs in practical, real-world settings.

## Limitations

Due to computational constraints, we do not evaluate our methods on the full CWQ and WebQSP datasets. Instead, following the setting of (Sun et al., 2024), we sample a subset of questions from each dataset for evaluation. While this approach may introduce greater variance in the results, the consistent performance trends observed across different models still provide strong evidence supporting our findings.

## Acknowledgments

We thank the reviewers and the ASUS AIoT team for their valuable feedback.

## References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Tu Ao, Yanhua Yu, Yuling Wang, Yang Deng, Zirui Guo, Liang Pang, Pinghui Wang, Tat-Seng Chua, Xiao Zhang, and Zhen Cai. 2025. Lightprof: A lightweight reasoning framework for large language model on knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(22):23424–23432.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Sitao Cheng, Ziyuan Zhuang, Yong Xu, Fangkai Yang, Chaoyun Zhang, Xiaoting Qin, Xiang Huang, Ling Chen, Qingwei Lin, Dongmei Zhang, Saravan Rajmohan, and Qi Zhang. 2024. Call me when necessary: LLMs can efficiently and faithfully reason over structured environments. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4275–4295, Bangkok, Thailand. Association for Computational Linguistics.

Google. 2024. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems*, volume 37, pages 132876–132907. Curran Associates, Inc.

Xujian Liang and Zhaoquan Gu. 2025. Fast think-on-graph: Wider, deeper and faster reasoning of large language model on knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24558–24566.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. 2024. Generate-on-graph: Treat LLM as both agent and KG for incomplete knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18410–18430, Miami, Florida, USA. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2412.15115*.

Zukang Yang, Zixuan Zhu, and Jennifer Zhu. 2025. CuriousLLM: Elevating multi-document question answering with LLM-enhanced knowledge graph reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 274–286, Albuquerque, New Mexico. Association for Computational Linguistics.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

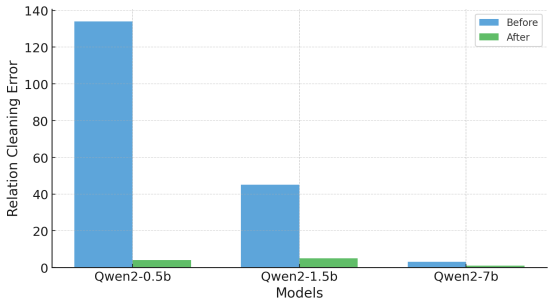



Figure 2: Relation cleaning errors before and after applying constrained decoding. Smaller models like Qwen2-0.5b and Qwen2-1.5b show substantial reductions in formatting errors, indicating the effectiveness of our constrained decoding strategy.

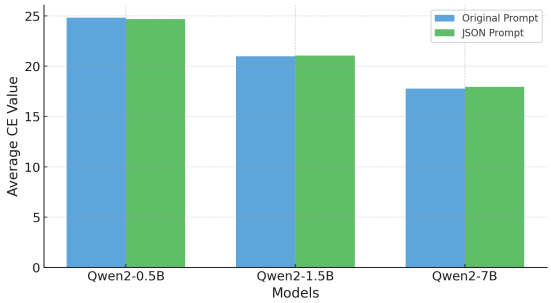



Figure 3: Average cross-entropy between model-retrieved relation paths and the pseudo-ground truth, before and after applying constrained decoding. The minimal differences suggest that constrained decoding does not compromise model exploration capability.

## A Implementation Details of Passage Retrieval for KG Exploration

Following the implementation of (Sun et al., 2024), our KG exploration framework adopts a lightweight retrieval module at each step to select relevant candidates from a predefined list. Given a question $q$, and a list of candidate passages $P_{cand}$ (either relation phrases or entity names), the goal of retrieval is to identify the top-$k$ most relevant candidates that guide the next reasoning step.

### Retrieval Formulation

For each step, we compute a relevance score between the question $q$ and every candidate passage $p \in P_{\text{cand}}$. The top-$k$ passages with the highest scores are selected:

$$P_q = \text{Top}_k \left(\text{score}(p, q)\right), \quad \forall p \in P_{\text{cand}}.$$

The scoring function $\text{score}(p, q)$ depends on the retrieval method used (BM25 or embedding-based retrievers).

### BM25 Retriever

For keyword-based retrieval, we use BM25 via the `rank_bm25` implementation. Each passage (e.g., a relation like "place of birth" or an entity name like "Albert Einstein") is treated as a short bag-of-words document. The question $q$ is tokenized into a word list $q_1, \cdots, q_n$, and its relevance to each passage is computed based on term frequency and inverse document frequency:

$$\text{score}(p, q) = \text{BM25}(p, q)$$

### Embedding-Based Retrievers

For embedding-based retrievers such as Sentence-BERT and GTR, we encode both the question and candidate passages using a pretrained text encoder $\mathcal{T}(\cdot)$. The relevance score is computed as the dot product between their embeddings:

$$\text{score}(p, q) = \langle \mathcal{T}(p), \mathcal{T}(q) \rangle.$$

## B Constrained Decoding with JSON Format

To ensure that the performance gap between SLMs and LLMs is not simply due to formatting inconsistencies or output mismatches, we adopt a constrained decoding strategy across all models. Specifically, we modify the prompts to require all models to produce answers strictly in a predefined JSON format. Comparisons of original prompt and our modified prompt are showed in Table 6 and 7.

By enforcing the constrained output structure, we ensure that all models, regardless of size, are evaluated under consistent conditions. We also conducted a quantitative analysis of relation cleaning errors before and after applying constrained decoding. Specifically, we counted how many times the model-generated outputs contained unparseable relation entries. As shown in Figure 2, constrained decoding substantially reduces relation formatting errors, especially for smaller models like Qwen2-0.5b and Qwen2-1.5b. This confirms that our constrained format enforcement effectively standardizes model outputs and mitigates noisy relation representations, allowing us to more reliably evaluate reasoning quality.

After removing parsing-related noise, we further examined whether the adoption of constrained decoding negatively impacts the LLMs' exploration ability. To assess this, we computed the cross entropy (CE) between the retrieved relation paths and

| Models | CWQ | WebQSP |
|---|---|---|
| GPT-4.1 | **0.575** | **0.810** |
| w/ BM25 | 0.525 | 0.745 |
| w/ SentenceBERT | 0.520 | 0.775 |
| w/ GTR | 0.505 | 0.805 |

Table 5: The performance of GPT-4.1 equipped with different exploration modules.

the ground-truth paths under both the original and constrained prompt settings.

As shown in Figure 3, the CE values remain stable across models, with negligible changes before and after applying constrained decoding. This result confirms that our constrained decoding strategy effectively removes parsing-related variance without diminishing the LLMs' ability to explore and select relevant paths.

## C Passage Retrieval for LLMs

In an ablation study conducted by Sun et al. (2024), they showed that using lightweight passage retrieval models for exploration significantly reduced the number of LLM calls from $2ND + D + 1$ to $D + 1$ where $D$, $N$ are the numbers of iterations and reasoning paths respectively. However, this efficiency gain came at the cost of a substantial drop in EM score. We reproduce the results in Table 5. In contrast, our experiments in Section 3.4 demonstrate that passage retrieval models can offer the best of both worlds for SLMs: not only do they improve the efficiency of ToG, but they also enhance the EM performance, without facing the trade-off observed in the original study. The main reason for this difference in findings lies in the disparity between LLMs and SLMs in their ability to perform KG exploration. Therefore, their results complement, rather than contradict our findings.

**Original Extract Relation Prompt (Unconstrained)**

Please retrieve 3 relations (separated by semicolon) that contribute to the question and rate their contribution on a scale from 0 to 1 (the sum of the scores of %s relations is 1).

Q: Name the president of the country whose main spoken language was Brahui in 1980?
Topic Entity: Brahui Language
Relations: language.human_language.main_country; language.human_language.language_family; language.human_language.iso_639_3_code; base.rosetta.languoid.parent; language.human_language.writing_system; base.rosetta.languoid.languoid_class; language.human_language.countries_spoken_in; kg.object_profile.prominent_type; base.rosetta.languoid.document; base.ontologies.ontology_instance.equivalent_instances; base.rosetta.languoid.local_name; language.human_language.region
A:
1. {language.human_language.main_country (Score: 0.4))}: This relation is highly relevant as it directly relates to the country whose president is being asked for, and the main country where Brahui language is spoken in 1980.
2. {language.human_language.countries_spoken_in (Score: 0.3)}: This relation is also relevant as it provides information on the countries where Brahui language is spoken, which could help narrow down the search for the president.
3. {base.rosetta.languoid.parent (Score: 0.2)}: This relation is less relevant but still provides some context on the language family to which Brahui belongs, which could be useful in understanding the linguistic and cultural background of the country in question.
Q:

**Modified Extract Relation Prompt (Constrained Decoding)**

Please retrieve 3 relations that contribute to the question and rate their contribution on a scale from 0 to 1 (the sum of the scores of 3 relations is 1). Provide the output in JSON format.

Q: Name the president of the country whose main spoken language was Brahui in 1980?
Topic Entity: Brahui Language
Relations: language.human_language.main_country; language.human_language.language_family; language.human_language.iso_639_3_code; base.rosetta.languoid.parent; language.human_language.writing_system; base.rosetta.languoid.languoid_class; language.human_language.countries_spoken_in; kg.object_profile.prominent_type; base.rosetta.languoid.document; base.ontologies.ontology_instance.equivalent_instances; base.rosetta.languoid.local_name; language.human_language.region
A:
```
{
  "relations": [
    {
      "relation": "language.human_language.main_country",
      "score": 0.4,
      "description": "This relation is highly relevant as it directly relates to the country whose president is being asked
          for, and the main country where Brahui language is spoken in 1980."
    },
    {
      "relation": "language.human_language.countries_spoken_in",
      "score": 0.3,
      "description": "This relation is also relevant as it provides information on the countries where Brahui language is
          spoken, which could help narrow down the search for the president."
    },
    {
      "relation": "base.rosetta.languoid.parent",
      "score": 0.2,
      "description": "This relation is less relevant but still provides some context on the language family to which
          Brahui belongs, which could be useful in understanding the linguistic and cultural background of the
          country in question."
    }
  ]
}
```
Q:

Table 6: Comparison of original prompt and our constrained decoding version for relation pruning. The modified prompt enforces a strict JSON structure to enable consistent and parseable outputs from SLMs.

**Original Score Entity Candidates Prompt (Unconstrained)**

lease score the entities' contribution to the question on a scale from 0 to 1 (the sum of the scores of all entities is 1).

Q: The movie featured Miley Cyrus and was produced by Tobin Armbrust?
Relation: film.producer.film
Entites: The Resident; So Undercover; Let Me In; Begin Again; The Quiet Ones; A Walk Among the Tombstones
Score: 0.0, 1.0, 0.0, 0.0, 0.0, 0.0
The movie that matches the given criteria is "So Undercover" with Miley Cyrus and produced by Tobin Armbrust. Therefore, the score for "So Undercover" would be 1, and the scores for all other entities would be 0.

Q: {}
Relation: {}
Entites:

**Modified Score Entity Candidates Prompt (Constrained Decoding)**

Please score each entity's contribution to the question on a scale from 0 to 1 (the sum of the scores of all entities should be 1). Provide the output in JSON format.

Q: The movie featured Miley Cyrus and was produced by Tobin Armbrust?
Relation: film.producer.film
Entities: The Resident; So Undercover; Let Me In; Begin Again; The Quiet Ones; A Walk Among the Tombstones

```
A: {{
  "entities": [
    {{"name": "The Resident", "score": 0.0}},
    {{"name": "So Undercover", "score": 1.0}},
    {{"name": "Let Me In", "score": 0.0}},
    {{"name": "Begin Again", "score": 0.0}},
    {{"name": "The Quiet Ones", "score": 0.0}},
    {{"name": "A Walk Among the Tombstones", "score": 0.0}}
  ],
  "explanation": "The movie that matches the given criteria is \"So Undercover,\" which features Miley Cyrus and was
      produced by Tobin Armbrust. Therefore, the score for \"So Undercover\" is 1, and the scores for all other
      entities are 0."
}}
```

Q: {}
Relation: {}
Entities:

Table 7: Comparison of original prompt and our constrained decoding version for entities pruning. The modified prompt enforces a strict JSON structure to enable consistent and parseable outputs from SLMs.