Think-On-Graph : Deep and Responsible Reasoning of Large Language Model on Knowledge Graph

Sun, Jiashuo, et al.
ICLR 2024

Presented by Hyemin Yang in the Data Mining Lab at Hanyang University

Table of contents

Part 1 Problem Statement

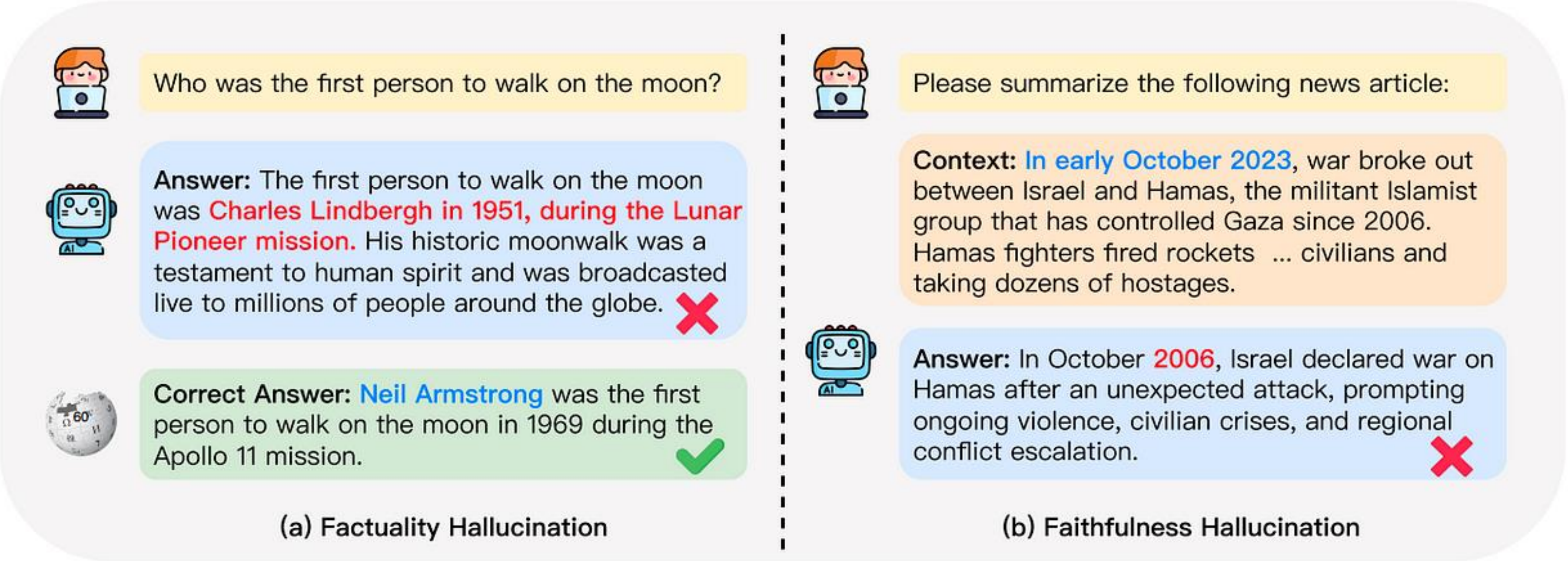
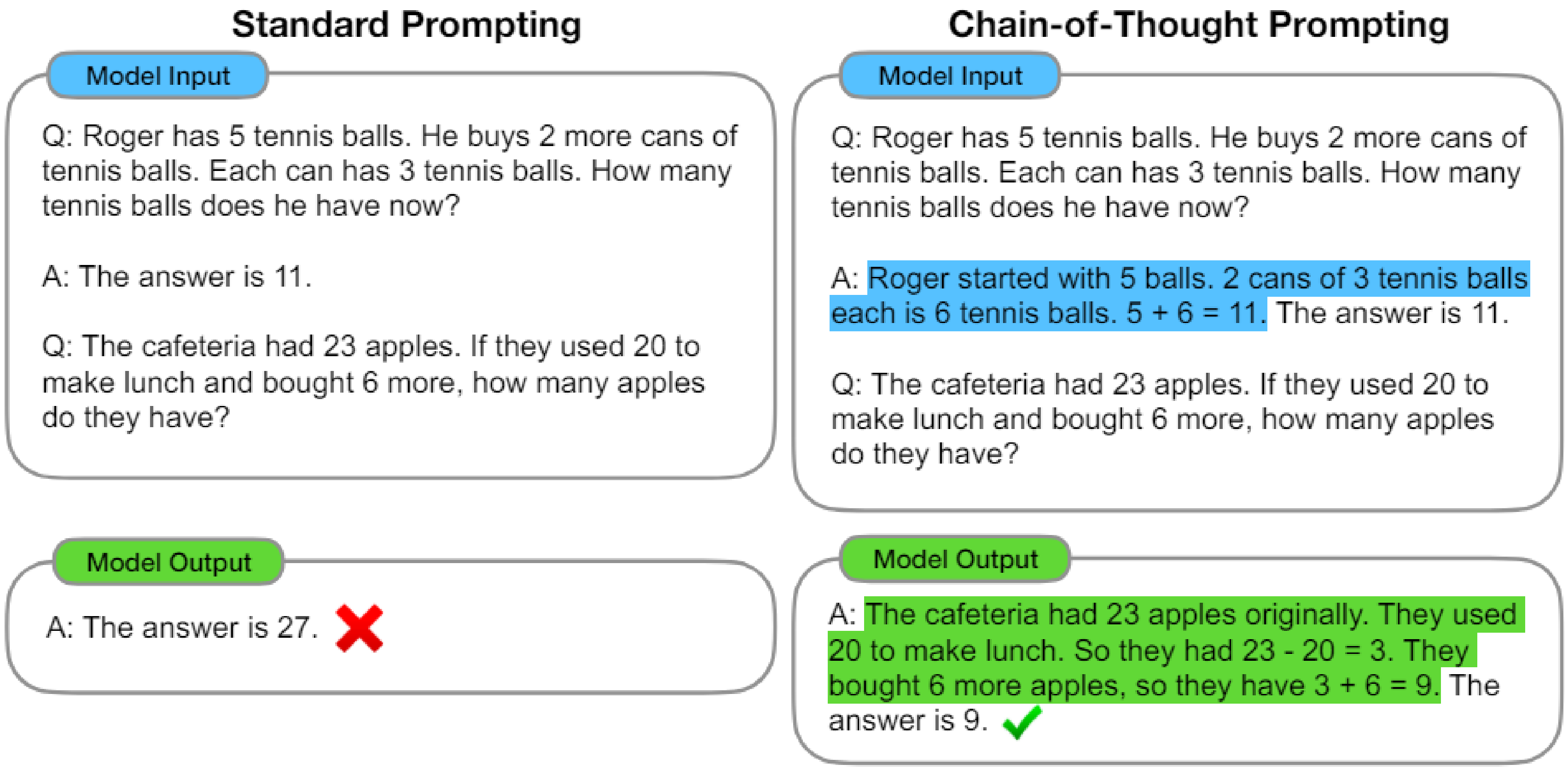


Figure 1: An intuitive example of LLM hallucination.

Part 1 Problem Statement



Problem Statement

Question:
`What is the majority party now in the country where Canberra is located?`

LLM-only

(Chain-of-Thought Prompt): Let's think step by step.
Response: Canberra is the capital of Australia. According to my knowledge up to September 2021 the prime minister of Australia is Scott Morrison, who is a member of the Liberal Party. So the answer should be **Liberal Party**. ❌

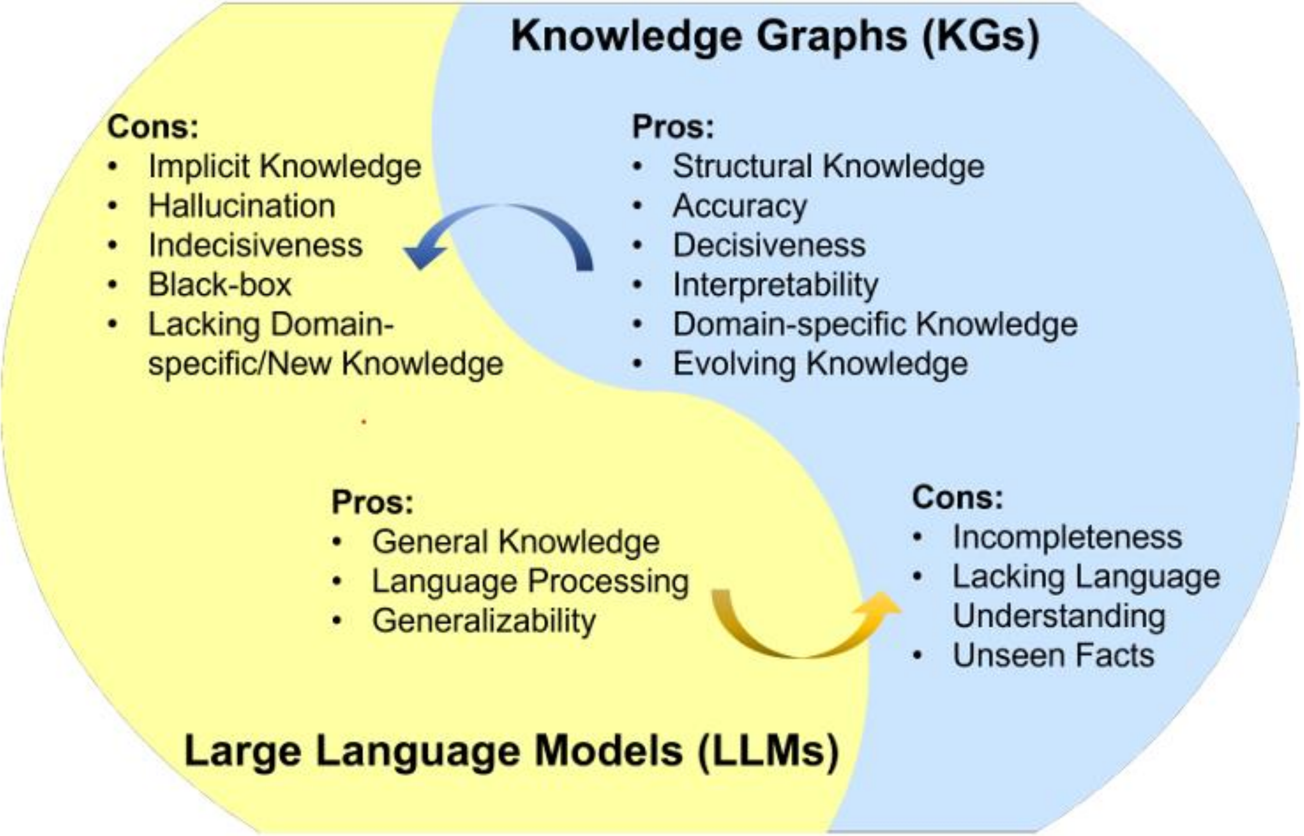
(a)

LLMs usually fail to provide accurate answers to

- questions requiring specialized knowledge beyond what was included in the pre-training phase (e.g., out-of-date knowledge)
- questions requiring long logic chain and multi-hop reasoning

The training process for LLMs is often expensive and time-consuming, making it challenging to keep their knowledge up to date!

Problem Statement



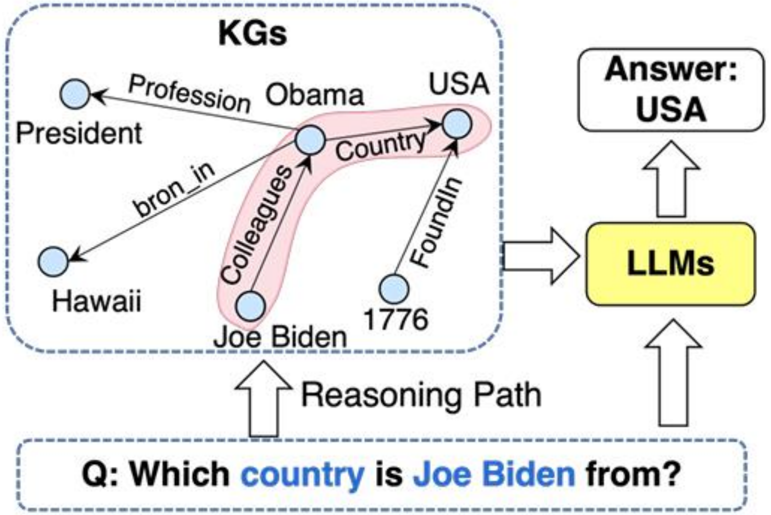
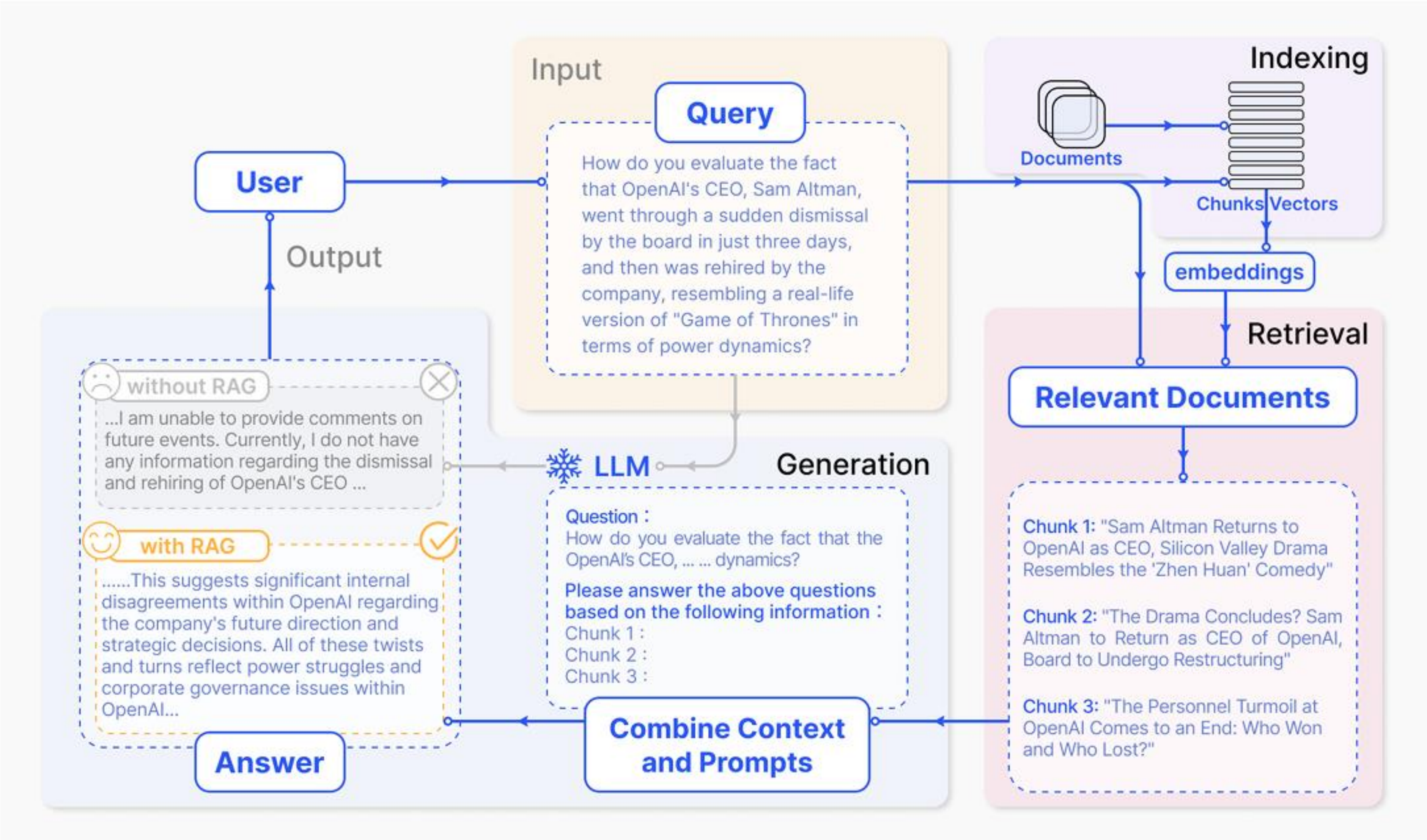


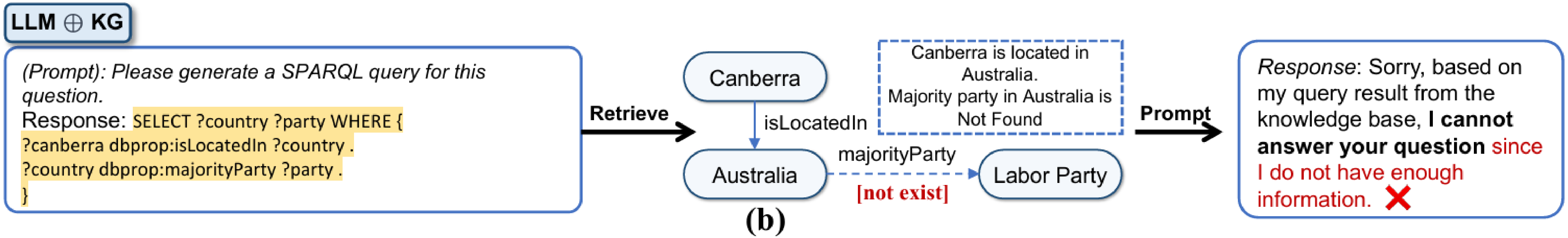
Fig. 13. The general framework of using knowledge graph for language model analysis.

Part 1 Problem Statement



Part 1 **Problem Statement**

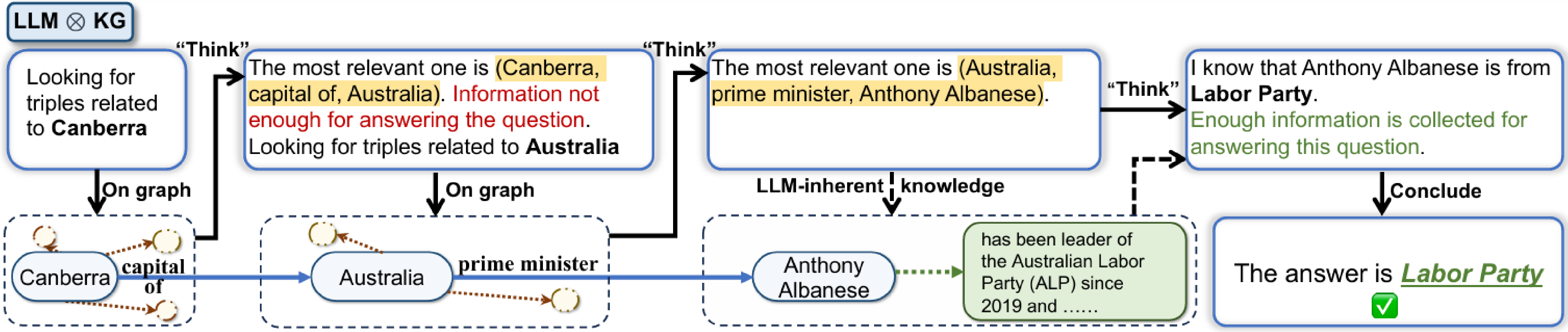
Question:
What is the majority party now in the country where Canberra is located?



Part 2

Method : "Think on Graph (ToG)"

Question:
What is the majority party now in the country where Canberra is located?



(c)

KGs and LLMs work in tandem in each step of graph reasoning

Method : Asking LLM to perform beam search on knowledge graph

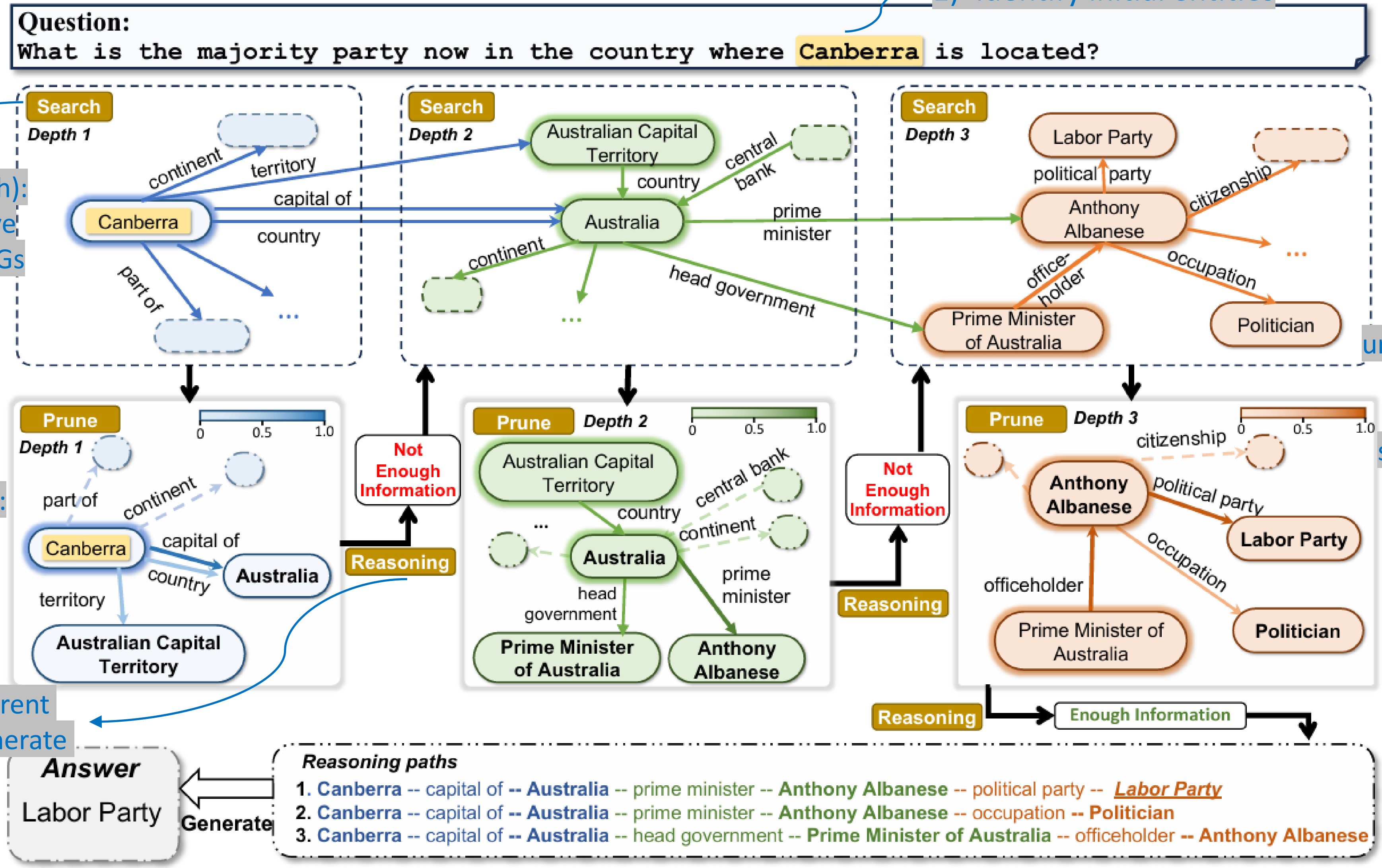
1) Identify initial entities

2-1) Exploration (search):
Call the LLM to retrieve relevant triples from KGs

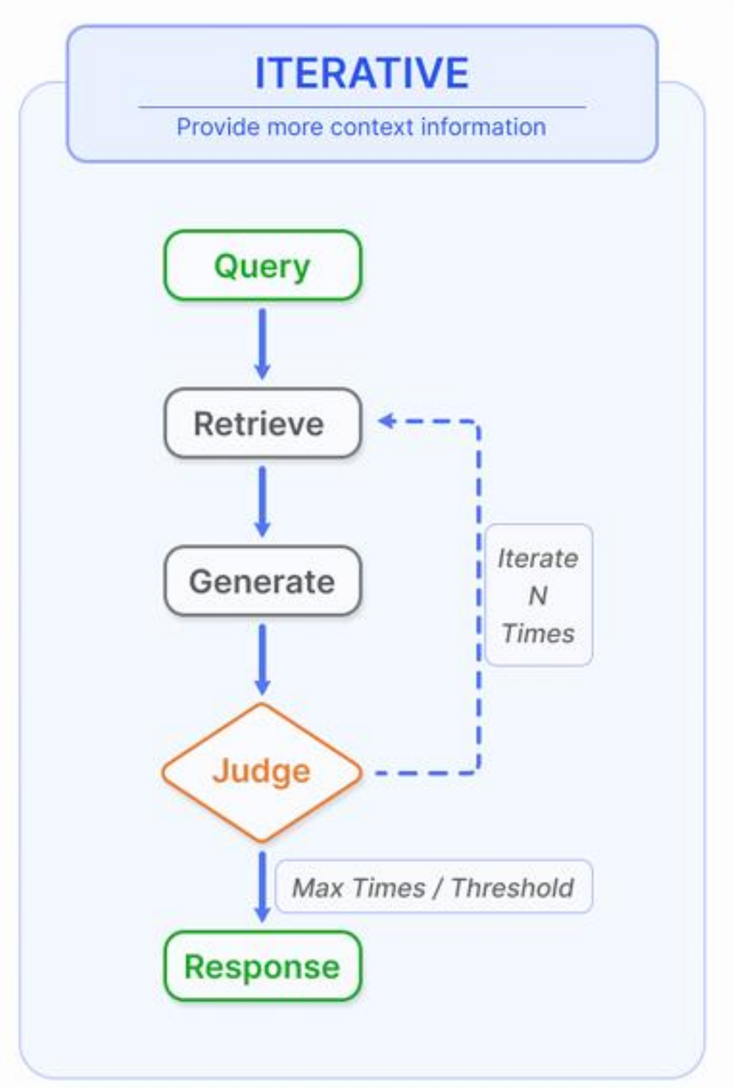
2-2) Exploration (prune):
Decide the most relevant triples

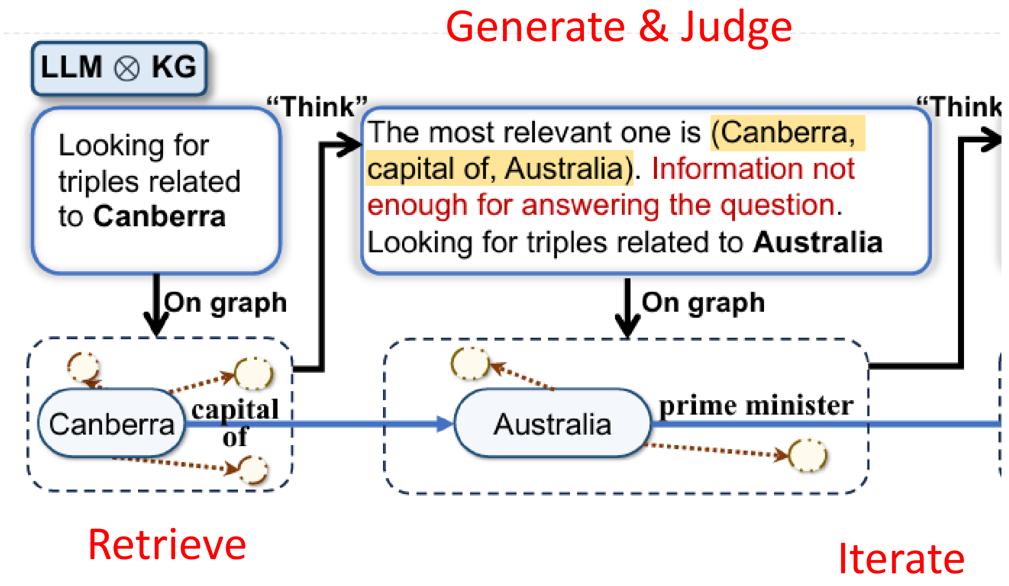
3) Reasoning:
Evaluate whether the current paths are adequate for generate the answer

3) Iterate 2,3
until adequate information is gathered or the predefined maximum search depth is reached.



Method : Asking LLM to perform beam search on knowledge graph



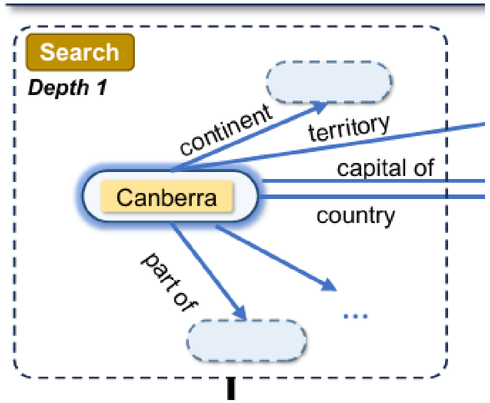


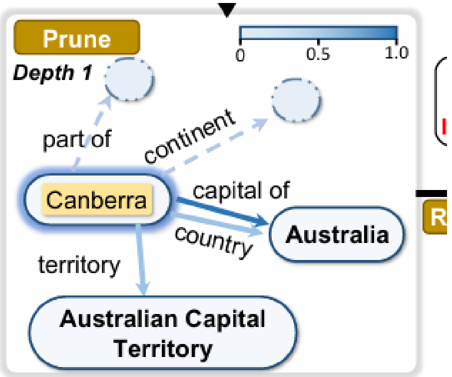
Method : The entire inference process of ToG

ToG constantly updates and maintains top-N reasoning paths $P = \{p_1, p_2, \ldots, p_N\}$ for the question x after each iteration.

1) Initialization :
   - The initialization of the top-N reasoning paths P.
   - The LLM automatically extract the topic entities in questions.
   - $E^0 = \{e_1^0, e_2^0, ..., e_N^0\}$

2) Exploration :
   - Relation Exploration → Entity Exploration → Relation Exploration → ...
   - Beam search process with the depth 1 and the width of N from E^{D-1} to R^D (R^D to E^D) at the D-th iteration.
   - Search and Prune.





Method : Exploration examples (Search)

E.1.1 Relation Search

```
PREFIX ns: <\protect\vrule width0pt\protect\href{http://rdf.freebase.com/
    ns/}{http://rdf.freebase.com/ns/}>
SELECT ?relation
WHERE {
    ns:mid ?relation ?x .
}
```

```
PREFIX ns: <\protect\vrule width0pt\protect\href{http://rdf.freebase.com/
    ns/}{http://rdf.freebase.com/ns/}>
SELECT ?relation
WHERE {
    ?x ?relation ns:mid .
}
```

E.1.2 Entity Search

```
PREFIX ns: <\protect\vrule width0pt\protect\href{http://rdf.freebase.com/
    ns/}{http://rdf.freebase.com/ns/}>
SELECT ?tailEntity
WHERE {
    ns:mid ns:relation ?tailEntity .
}
```

```
PREFIX ns: <\protect\vrule width0pt\protect\href{http://rdf.freebase.com/
    ns/}{http://rdf.freebase.com/ns/}>
SELECT ?tailEntity
WHERE {
    ?tailEntity ns:mid ns:relation  .
}
```

Method : Exploration examples (Prune)

E.3.1 RELATION PRUNE

Please retrieve k 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 k relations is 1).

`In-Context Few-shot`

Q: {Query}

Topic Entity: {Topic Entity}

Relations: {list of relations}

A:

E.3.2 ENTITY PRUNE

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

`In-Context Few-shot`

Q: {Query}

Relation: {Current Relation}

Entites: {list of entities}

Score:

Part 2 Method : The entire inference process of ToG

3) Reasoning :
- The LLM evaluates whether the current reasoning paths are adequate for generating the answer.
- If the evaluation yields a positive result, we prompt the LLM to generate the answer using the reasoning paths with the query as inputs.
- If the evaluation yields a negative result, we repeat 'Exploration' and 'Reasoning' steps until the evaluation is positive or reaches the maximum depth.

E.3.3 REASONING

Given a question and the associated retrieved knowledge graph triples (entity, relation, entity), you are asked to answer whether it's sufficient for you to answer the question with these triples and your knowledge (Yes or No).

`In-Context Few-shot`

Q: {Query}

Knowledge triples: {Explored Paths}

A:

Method : Reasoning examples

E.3.4 GENERATE

Given a question and the associated retrieved knowledge graph triples (entity, relation, entity), you are asked to answer the question with these triples and your own knowledge.

```
In-Context Few-shot
```

Q: {Query}

Knowledge triples: {Explored Paths}

A:

- ToG needs at most $2ND + D + 1$ calls to the LLM.

Part 3 **Experiments : Main results**

Method	Multi-Hop KBQA				Single-Hop KBQA	Open-Domain QA	Slot Filling		Fact Checking
	CWQ	WebQSP	GrailQA	QALD10-en	Simple Questions	WebQuestions	T-REx	Zero-Shot RE	Creak
Without external knowledge									
IO prompt w/ChatGPT	37.6	63.3	29.4	42.0	20.0	48.7	33.6	27.7	89.7
CoT w/ChatGPT	38.8	62.2	28.1	42.9	20.3	48.5	32.0	28.8	90.1
SC w/ChatGPT	45.4	61.1	29.6	45.3	18.9	50.3	41.8	45.4	90.8
With external knowledge									
Prior FT SOTA	70.4[α]	82.1[β]	75.4[γ]	45.4[δ]	85.8[ε]	56.3[ζ]	87.7[η]	74.6[θ]	88.2[ι]
Prior Prompting SOTA	-	74.4[κ]	53.2[κ]	-	-	-	-	-	-
ToG-R (Ours) w/ChatGPT	58.9	75.8	56.4	48.6	45.4	53.2	75.3	86.5	93.8
ToG (Ours) w/ChatGPT	57.1	76.2	68.7	50.2	53.6	54.5	76.8	88.0	91.2
ToG-R (Ours) w/GPT-4	**69.5**	81.9	80.3	**54.7**	58.6	57.1	75.5	86.9	95.4
ToG (Ours) w/GPT-4	67.6	**82.6**	**81.4**	53.8	**66.7**	**57.9**	**77.1**	**88.3**	**95.6**

- ToG-R : relation-based ToG that explores the top-N relation chains $\{p_n = (e_n^0, r_n^1, r_n^2, ..., r_n^D)\}_{n=1}^{N}$ instead of triple-based reasoning paths. (ND + D + 1 calls to the LLM.)

Experiments : 1) Performance with different backbone models, 2) Ablation study

Method	CWQ	WebQSP
Fine-tuned		
NSM (He et al., 2021)	53.9	74.3
CBR-KBQA (Das et al., 2021)	67.1	-
TIARA (Shu et al., 2022)	-	75.2
DeCAF (Yu et al., 2023)	70.4	82.1
Prompting		
KD-CoT (Wang et al., 2023b)	50.5	73.7
StructGPT (Jiang et al., 2023)	-	72.6
KB-BINDER (Li et al., 2023a)	-	74.4
LLama2-70B-Chat		
CoT	39.1	57.4
ToG-R	**57.6**	**68.9**
ToG	53.6	63.7
Gain	(+18.5)	(+11.5)
ChatGPT		
CoT	38.8	62.2
ToG-R	57.1	75.8
ToG	**58.9**	**76.2**
Gain	(+20.1)	(+14.0)
GPT-4		
CoT	46.0	67.3
ToG-R	67.6	81.9
ToG	**69.5**	**82.6**
Gain	(+23.5)	(+15.3)

Table 2: Performances of ToG using different backbone models on CWQ and WebQSP.

Method	CWQ	WebQSP
ToG		
w/BM25	51.4	58.7
w/SentenceBERT	51.7	66.3
w/ChatGPT	58.8	76.2
ToG-R		
w/BM25	49.4	57.3
w/SentenceBERT	50.1	60.1
w/ChatGPT	59.2	75.1

Table 5: Performances of ToG using different pruning tools.

Experiments : Do search depth and width matter for ToG?

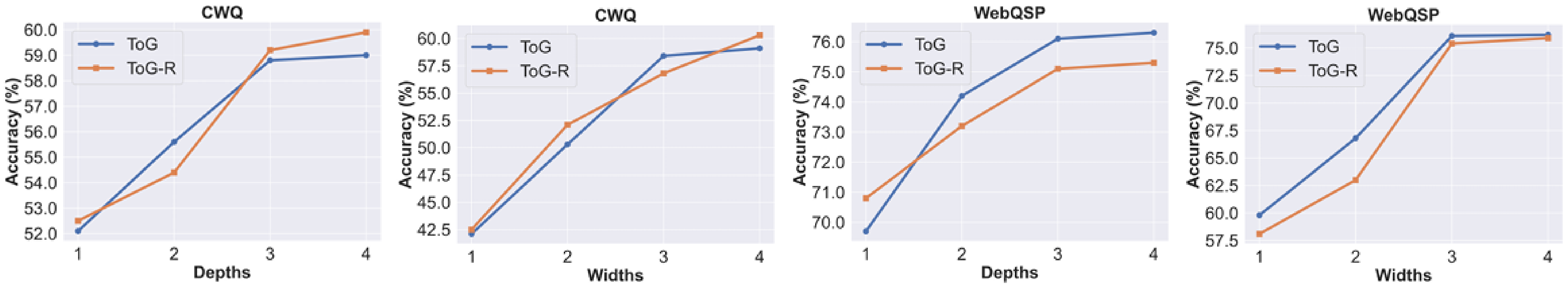


Figure 3: Performances of ToG with different search depths and widths.

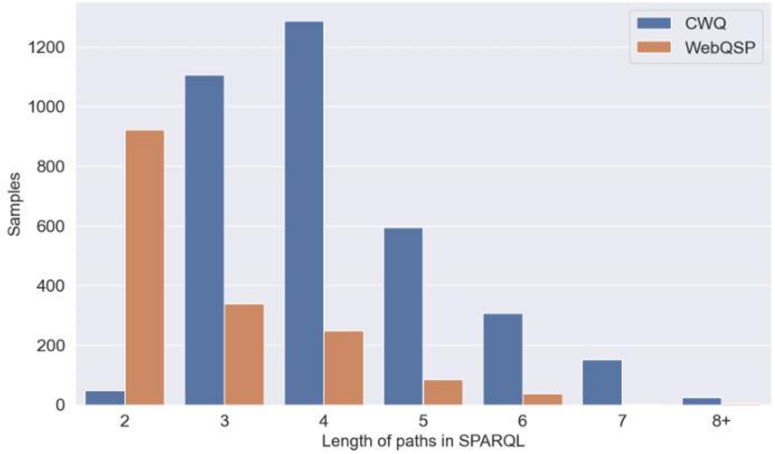


Figure 12: The lengths of the ground-truth SPARQL queries within the CWQ and WebQSP datasets, computed based on relation numbers.

Experiments : Result analysis

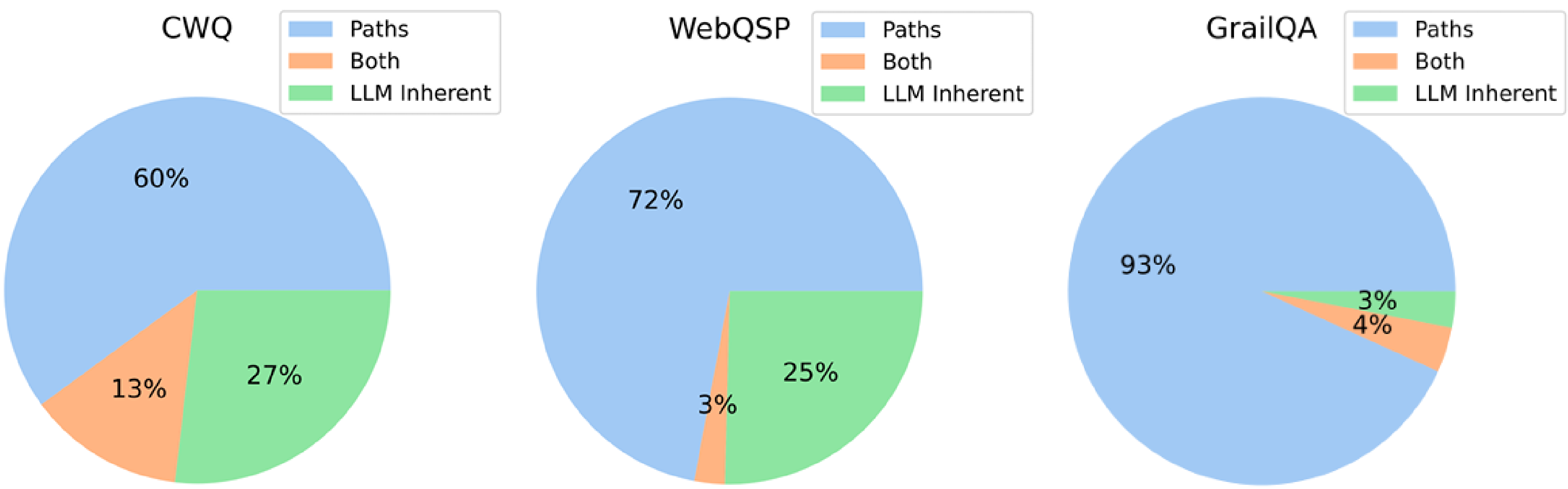


Figure 7: The proportions of ToG's evidence of answers on CWQ, WebQSP, and GrailQA datasets.

Part 3 Experiments : ToG can correct and construct KG.

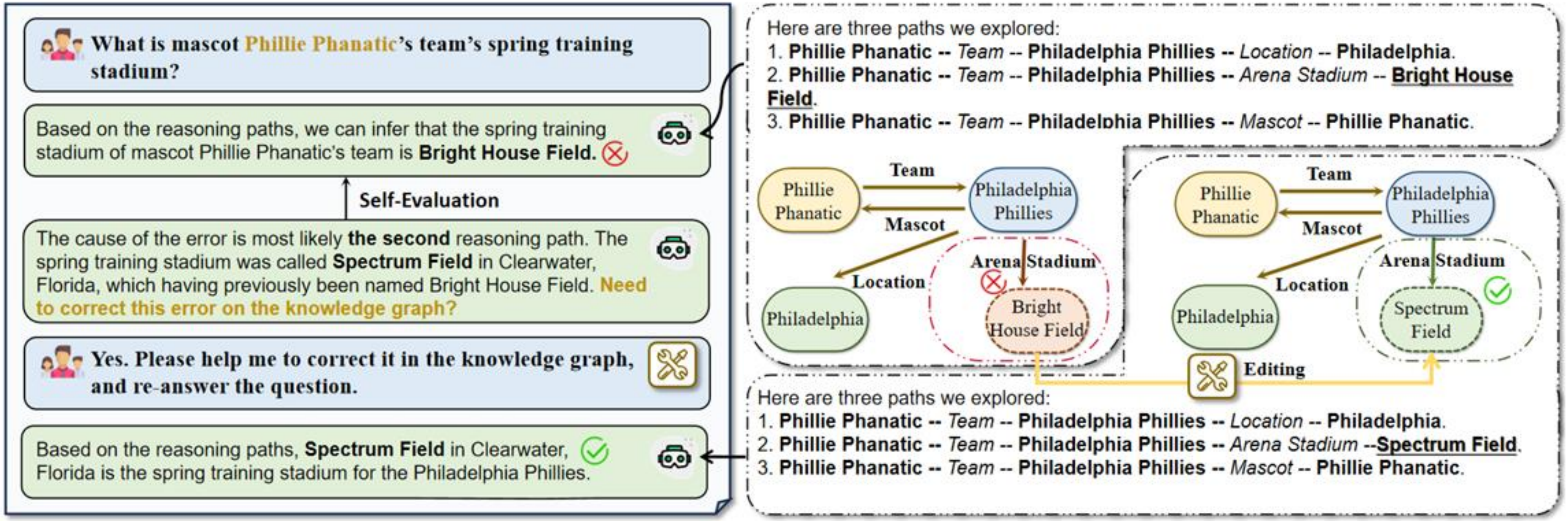


Figure 4: The illustration of knowledge traceability and correctability of ToG.

Part 4 Conclusion

- Limitations
  - Strong reliance on LLMs
  - Requires more LLM calls during reasoning
  - There are no results on small LLMs.

- Contributions
  - Provides new LLM+KG methods
  - Makes full use of the capabilities of the LLM