Symbolic Knowledge Injection in LLMs for Zero-Shot and Few-Shot Scenarios

Intern: Valentin Six

Advisors: Gaël de Chalendar, Evan Dufraisse

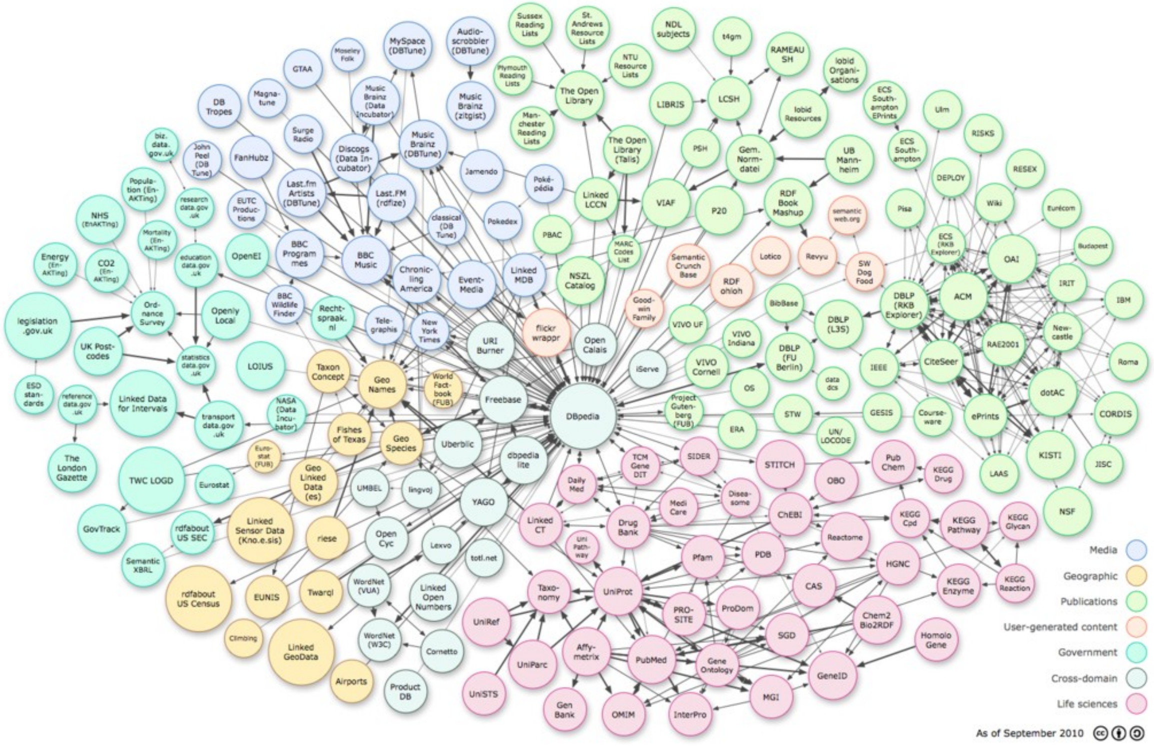
Table of content

Research Subject

- LLMs can lack explicit understanding and reasoning capabilities

- For certain tasks, data can be rare (or absent): training becomes difficult or expensive…

  → Objective: use **knowledge graphs** and exploit richness of **structure** to improve reasoning capacity of LLMs + ground the model with facts

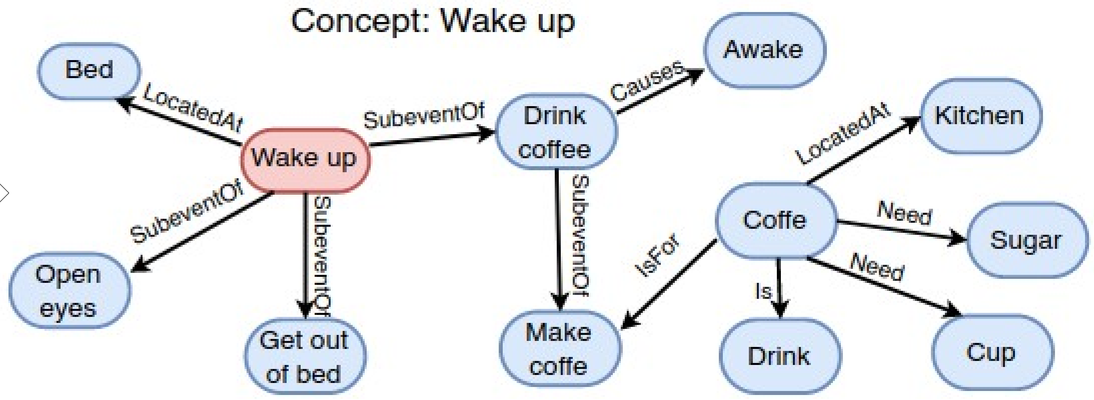


https://accidental-taxonomist.blogspot.com/2019/05/knowledge-graphs-and-ontologies.html

Knowledge Injection

What is a logical thing to do when waking up ?

Prompt



Pan et al. (2024)

You open your eyes and drink coffee

Output

→ Enable better **question understanding** + **reduce hallucinations**

→ Inject facts / implicit relationships (not mentioned in the prompt)

Knowledge Injection

Why using **knowledge graphs** ?

→ Density of information + rich structure

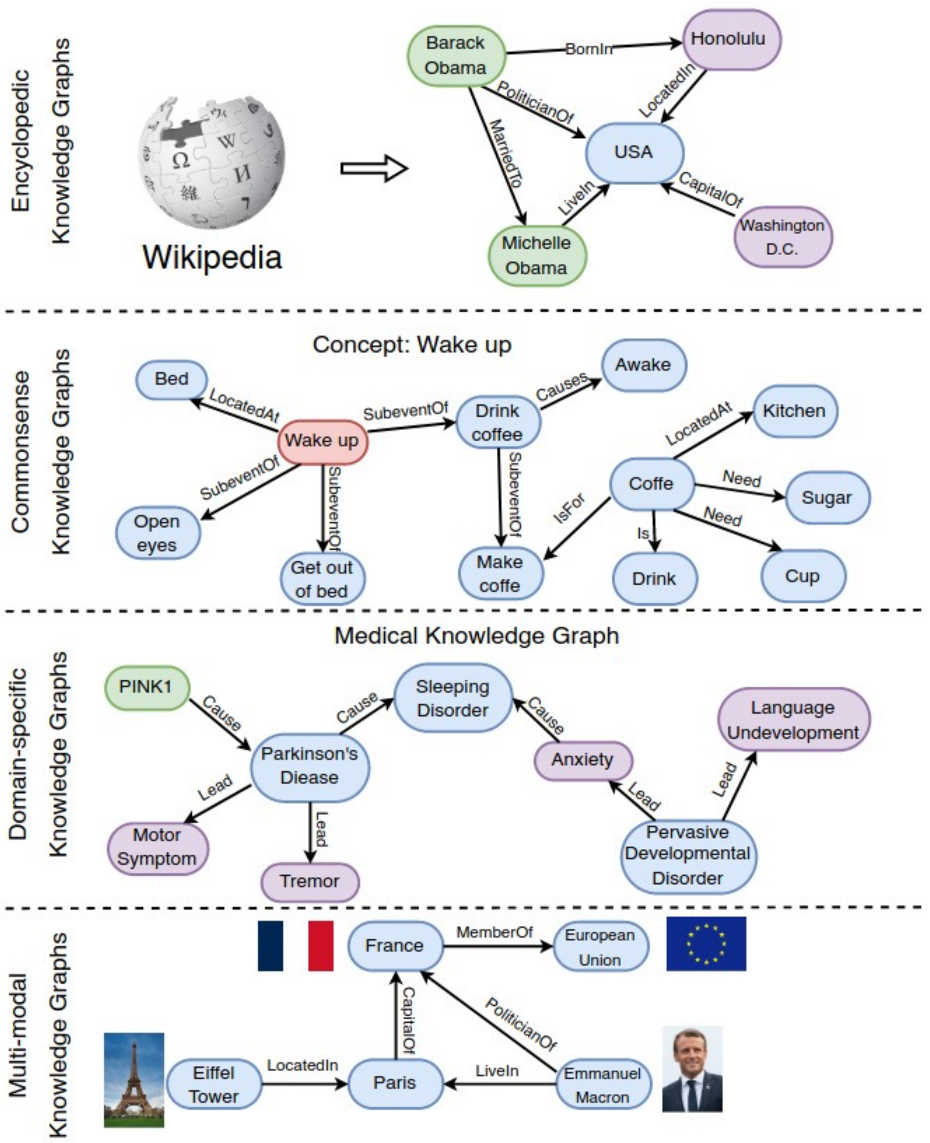
→ Traceability + results interpretation

→ Different graph types for different use cases

How to use them ?

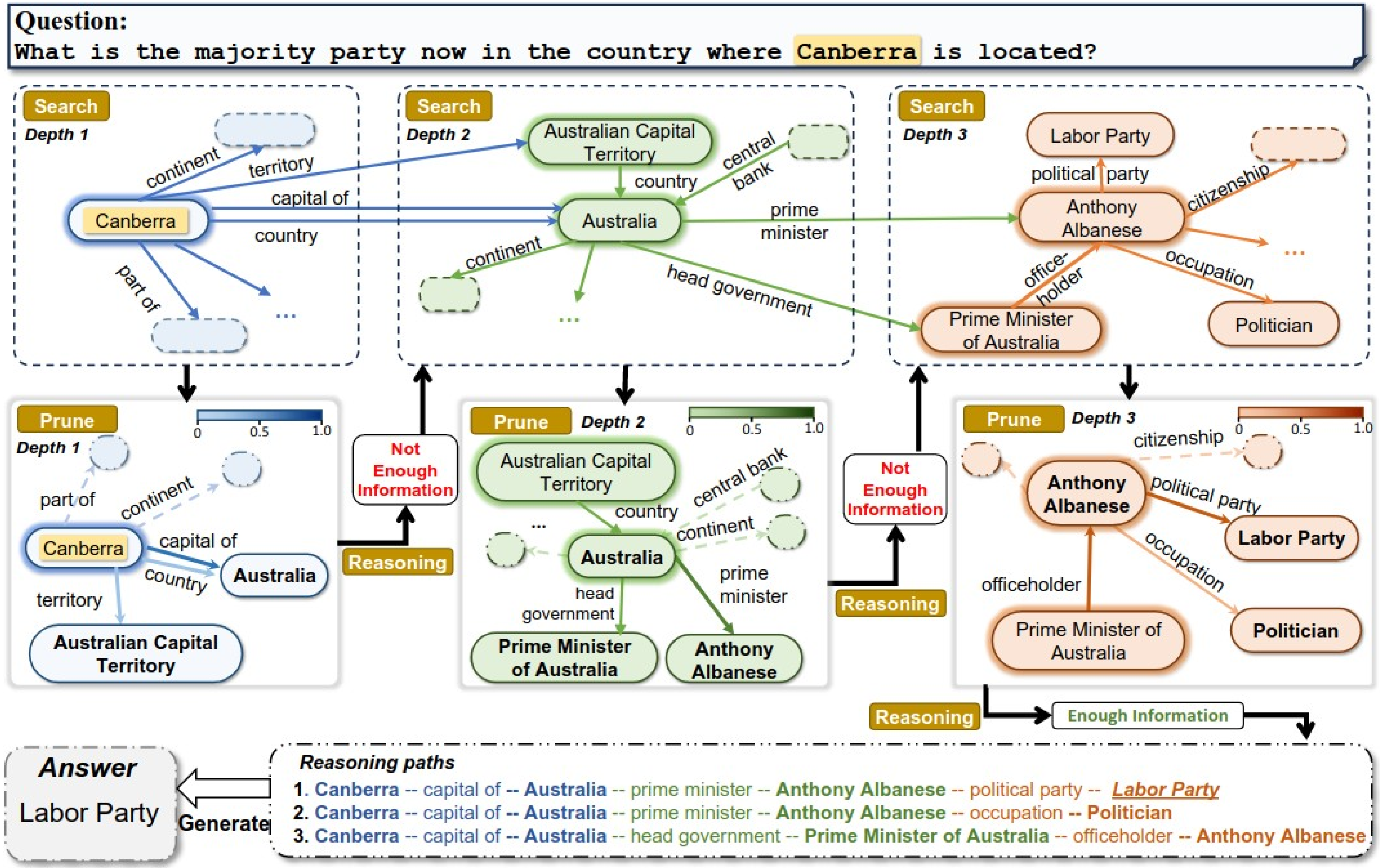
→ Triplets, reasoning paths, sub-graphs

→ Methods : GraphRAG, fine-tuning, pre-training



Pan et al. (2024)

How about reasoning ?



Sun et al. (2024)

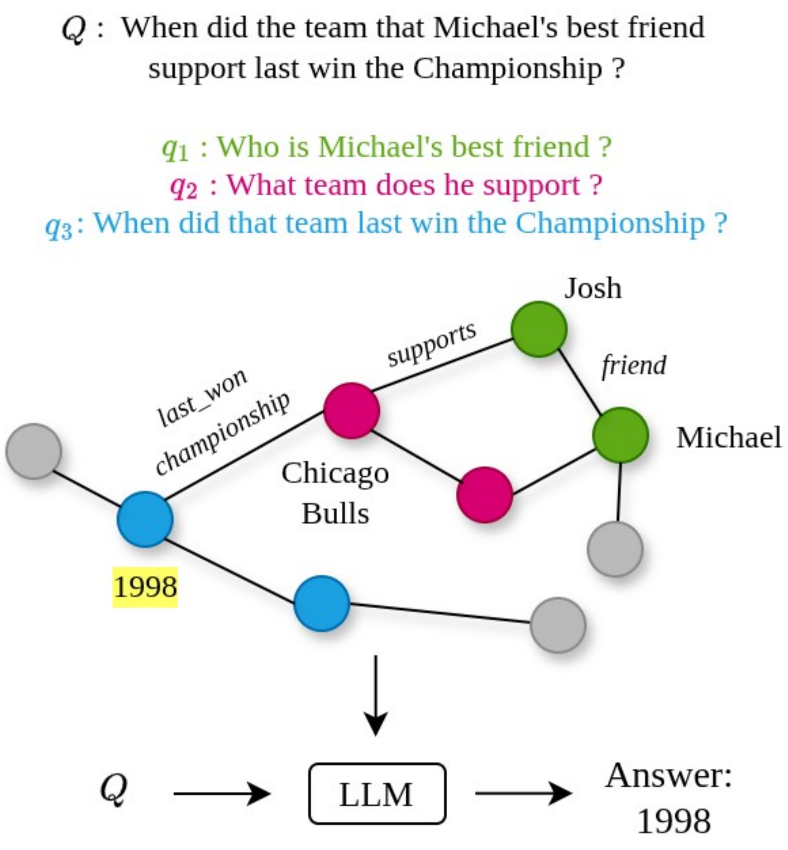
How to improve reasoning ?

Graph retrieval is a difficult task + expensive task for complex queries

Intuition :

- **Decomposing** complex questions in simple sub-questions and perform multiple retrievals
- Fully capture **all dimensions** of the complex question
- Use **reasoning** used within decomposition to improve graph retrieval

Using "G-Retriever" approach to handle textual graphs.



WebQSP vs. CWQ Benchmarks

- **WebQSP**: QA dataset, based on Freebase ; simple questions (2-hop max)

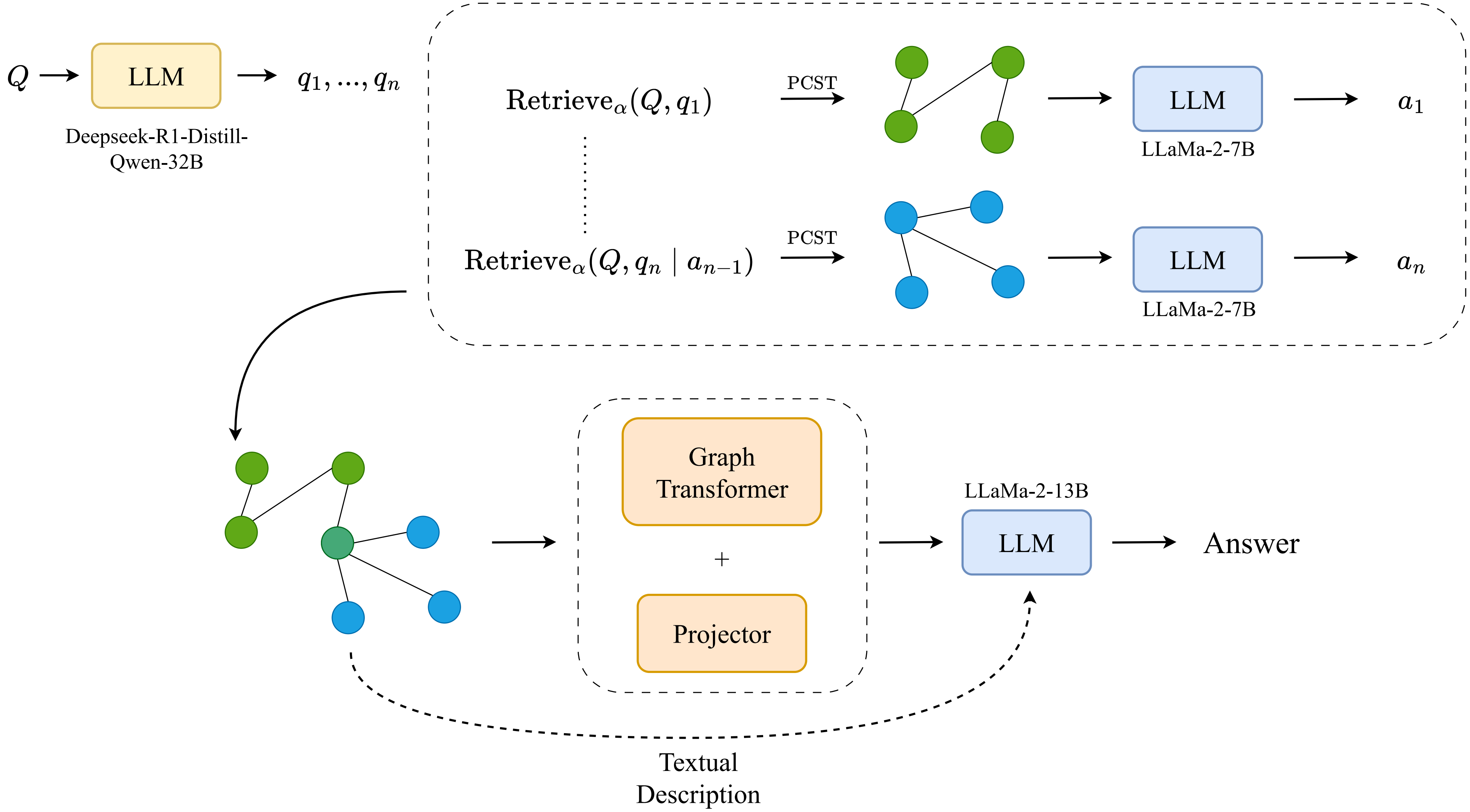
  → *used by G-Retriever*

- **CWQ** : also based on Freebase ; more complex questions (multi-hop)

  → *ignored by G-Retriever ; would probably struggle*

Objective: answer more <u>complex questions</u> that require <u>multiple steps + reasoning</u>

Overall Generation Pipeline



Retrieval and Graph Construction

1) Knowledge retrieval for each sub-question

$$\left.\begin{aligned} V_k^{\,i} &= \underset{n \in V}{argtopk}\left[\alpha \cos\left(z_{q_i}, z_n\right) + (1-\alpha)\cos\left(z_q, z_n\right)\right] \\ E_k^{\,i} &= \underset{e \in E}{argtopk}\left[\alpha \cos\left(z_{q_i}, z_e\right) + (1-\alpha)\cos\left(z_q, z_e\right)\right] \end{aligned}\right\}$$

Construct sub-graph $G_i = \left(V_k^{\,i}, E_k^{\,i}\right)$

Problem: sub-questions lack self-awareness ; not capturing the “global objective”

→ Retrieve the $\boldsymbol{k}$ most similar nodes + edges for each sub-question

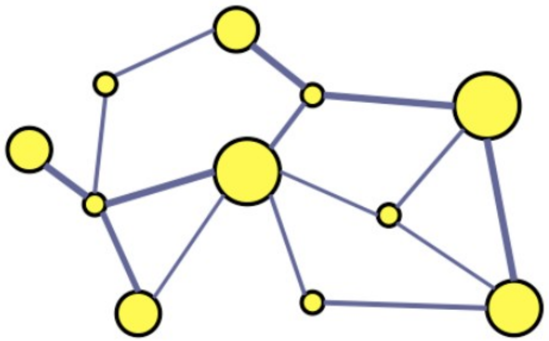
→ Using the initial question $\boldsymbol{Q}$ and current sub-question $\boldsymbol{q_i}$

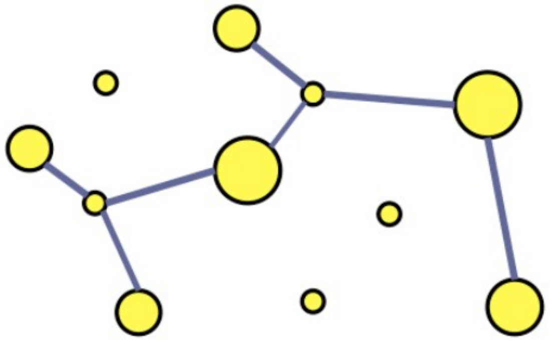
Parameter $\boldsymbol{\alpha}$ to control grounding ; “double cosine similarity”

Retrieval and Graph Construction

2) Graph transformation for each sub-question

→ Prize-Collecting Steiner Tree Algorithm (*TL;DR : most useful graph with minimal size)*





$$G_i = \left(V_k{}^i, E_k{}^i\right) \longrightarrow G_i{}^* = PCST\left(V_k{}^i, E_k{}^i\right)$$

Retrieved Graph

Filtered + Connected Graph

Akhmedov et al. (2018)

Retrieval and Graph Construction

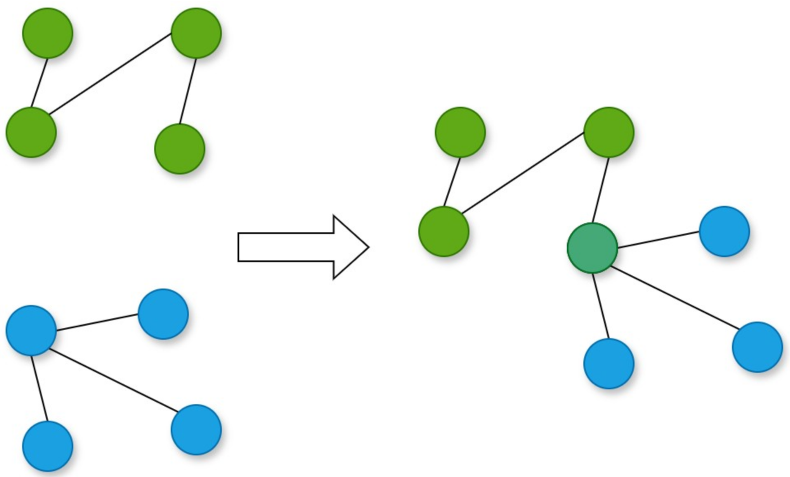
3) Graphs merging

$$G_i^* = PCST\left(V_k^{\,i}, E_k^{\,i}\right)$$

Obtaining connected sub-graphs

$$G^* = \coprod_{i}^{n} \left(G_1^*, \dots, G_n^*\right)$$

Merging



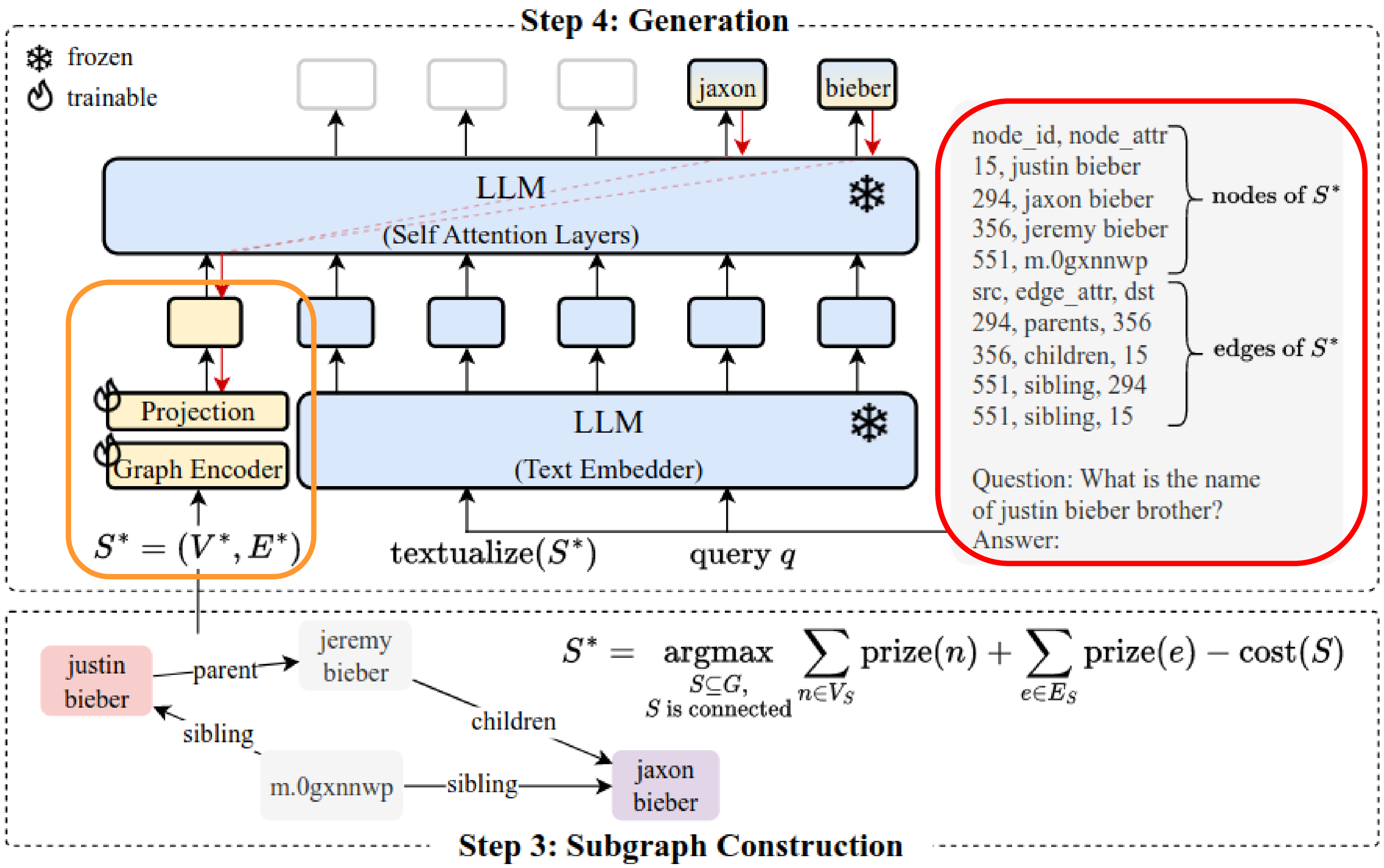
→ For each question, we **merge** the sub-graphs corresponding to the sub-questions

→ Detailed merging process:

- Extract sets of unique nodes + edges
- Create merged graph

Note: there is no guaranty that the final graph is connected, but most likely it will be

Answer Generation



He et al. (2024)

Generation:

Hard prompt + Soft prompt

Reconstruction:

Connected sub-graph from retrieved elements

Results

Method	CWQ		WebQSP	
	Hit@1	F1	Hit@1	F1
IO prompt (ChatGPT)	37.6	-	63.3	-
CoT (ChatGPT)	38.8	-	62.2	-
StructGPT (ChatGPT)	54.3	-	72.6	-
ToG (LLaMa-2-70B)	53.6	-	63.7	-
ToG (ChatGPT)	57.1	-	76.2	-
RoG (LLaMa-2-7B + FT)	<u>62.6</u>	56.2	**85.7**	70.8
PoG (GPT-3.5)	**63.2**	-	<u>82</u>	-
G-R (LLaMa-2-7B)	52.1	44.8	70.5	51.7
Ours (LLaMa-2-7B)	54.9	46	71.9	52.4
G-R (LLaMa-2-13B)	54.6	46.9	76.5	<u>57.2</u>
Ours (Hybrid 7B/13B)	<u>57.9</u>	<u>50.3</u>	**77.9**	**58.2**
Ours (LLaMa-2-13B)	**58.1**	**50.8**	<u>77.4</u>	56.4

Accuracy results compared to SOTA

Smaller models

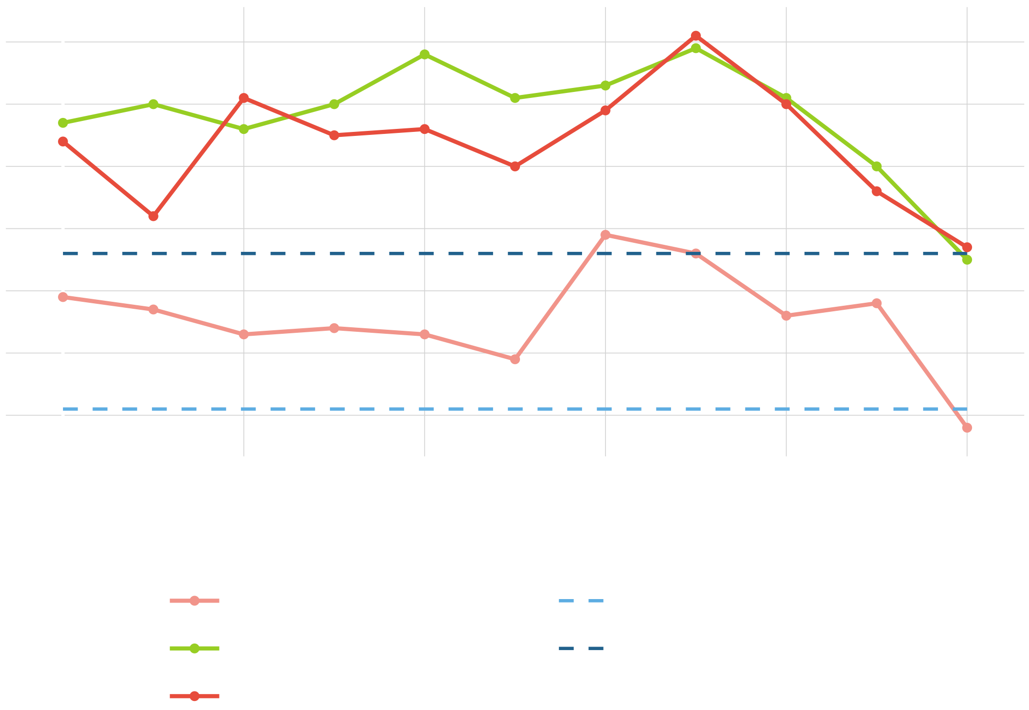
Larger models ≠ better performance

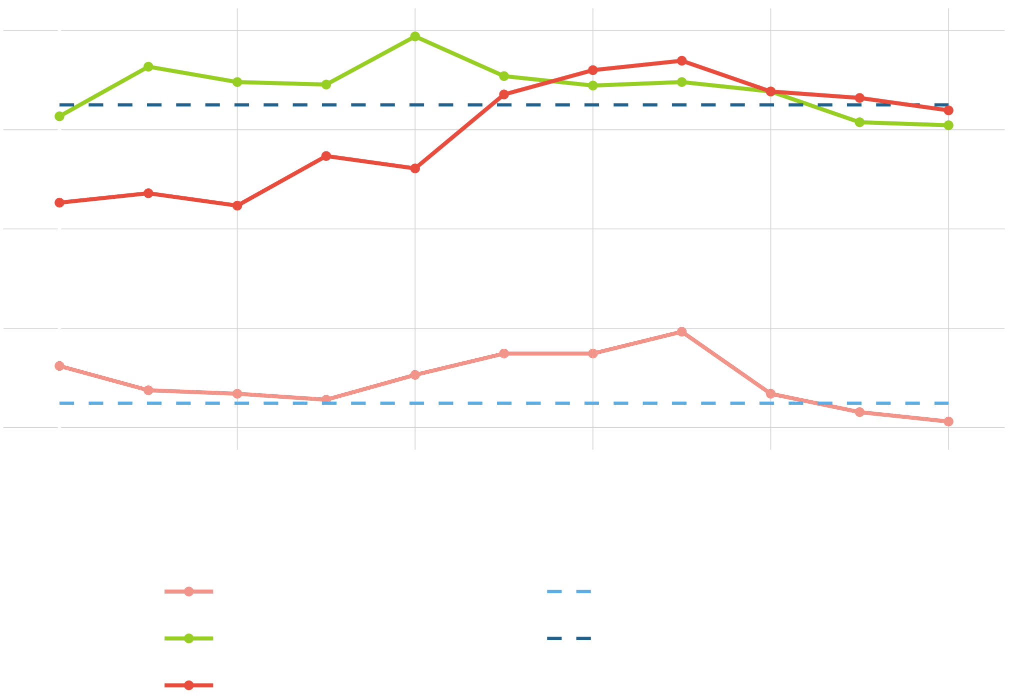
Fewer LLM calls

Method	CWQ	WebQSP
ToG	22.6	15.9
PoG	13.3	9.0
Ours	**4.8**	**4.3**

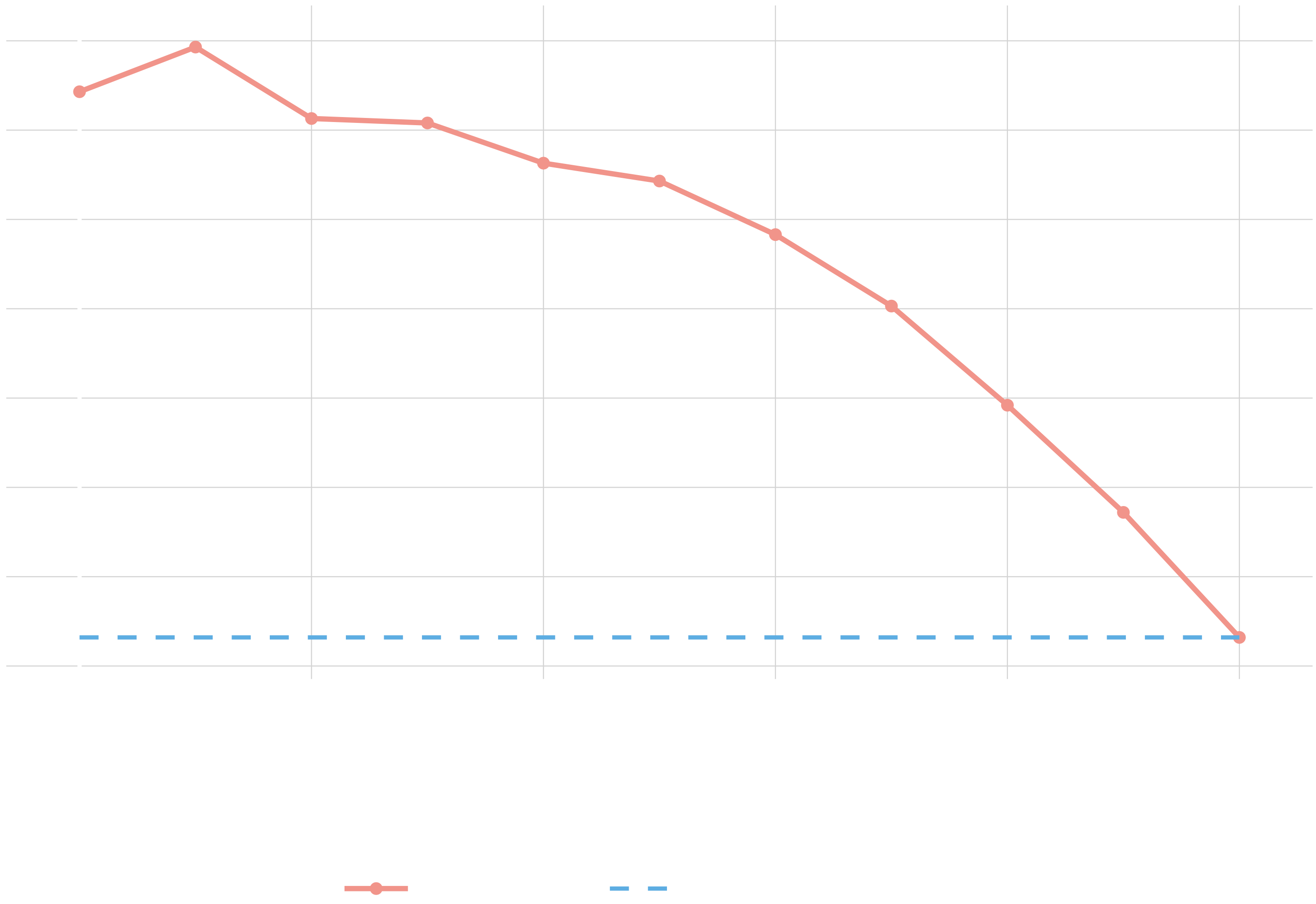
Number of LLM calls

Results





Results



References

« Unifying Large Language Models and Knowledge Graphs: A Roadmap » (Pan et. al)

« Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph » (Sun et al.)

« G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering » (He et al.)