LEGO-GraphRAG: Modularizing Graph-based Retrieval-Augmented Generation for Design Space Exploration

Yukun Cao, Zengyi Gao, Zhiyang Li, Xike Xie, S. Kevin Zhou, Jianliang Xu

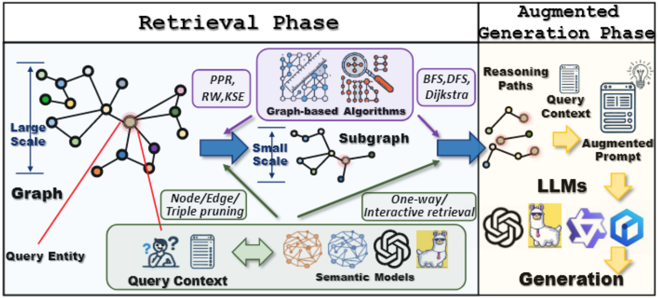
VLDB 2025

November 16, 2025

Outline

GraphRAG

- Retrieval-Augmented Generation (RAG) enhances LLMs with external knowledge.
- GraphRAG improves RAG by leveraging graph structures: entities, relationships, communities.



Challenges

- Lack of Unified Standards: different graph algorithms and semantic models.
- Difficulty in Modular Optimization: retrieval process is offen written as a monolithic block.
- Lack of Testing Platform: There is no public framework that can easily generate instances and perform large-scale comparisons.

Contributions

- LEGO-GraphRAG framework:
  - dividing the retrieval phase into two flexible modules: subgraph-extraction and path-retrieval.
  - classifies the techniques into structure-based and semantic-augmented methods for each module.
- supports implementing all existing GraphRAG instances.

LEGO-GraphRAG Architecture

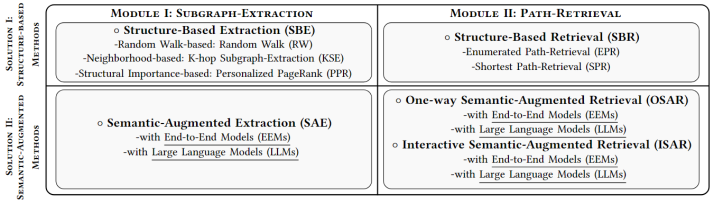


Figure: LEGO-GraphRAG Framework

- Two phase: Subgraph-Extraction; Path-Retrieval;
- Two method: Structure-based; Semantic-augmented;

Stucture-Based Extraction(SBE)

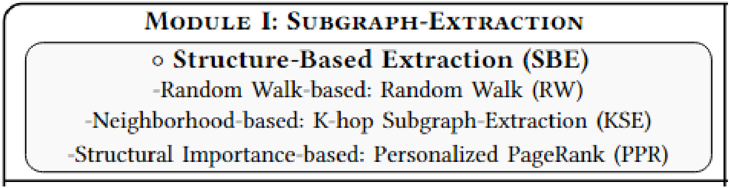


Figure: Stucture-Based Extraction

- Random Walk: randomly selecting edges and nodes at each step.
- K-hop Subgraph-Extraction: distance $d(v_i^q, v_j) \leq K$.
- Personalized PageRank: assign an importance score to each node.

Semantic-Augmented Extraction (SAE)

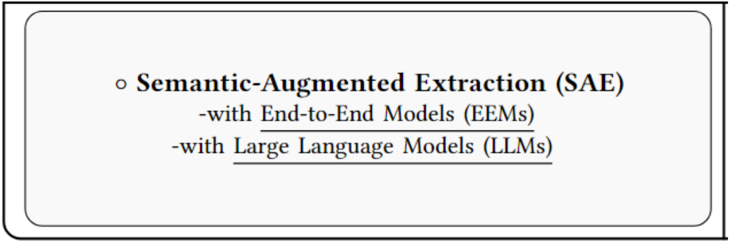


Figure: Semantic-Augmented Extraction

- With EEMs: compute semantic relevance.
  - pre-filtering: use SBE extracts a smaller subgraph.
  - subgraph pruning: node pruning, edge pruning, and triple pruning.
- With LLMs: evaluate and filter the semantic relevance of the initially extracted subgraph.

Structure-Based Retrieval (SBR)

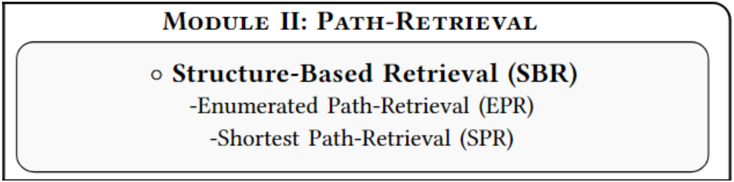


Figure: Structure-Based Retrieval

- Enumerated Path Retrieval: enumerates all possible paths from an entity $v_i^{(q)} \in \epsilon_q$.
- Shortest Path Retrieval: identifies all shortest paths from an entity $v_i^{(q)} \in \epsilon_q$.

One-way/Interactive Semantic-Augmented Retrieval (OSAR/ISAR)

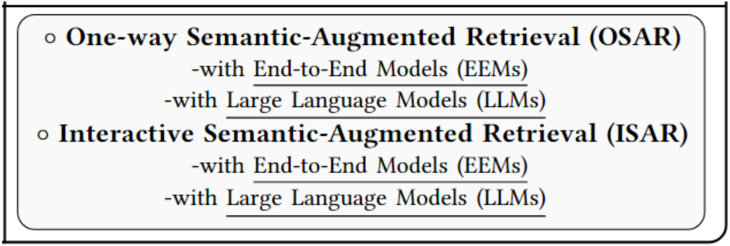


Figure: One-way/Interactive Semantic-Augmented Retrieval

- One-way Semantic-Augmented Retrieval: evaluate and select the N_p most relevant paths.
- Interactive Semantic-Augmented Retrieval: at each step, potential path extensions are evaluated for semantic relevance.

GraphRAG instances

Table 2: Five Groups of Instances under the LEGO-GraphRAG Framework

Group	Subgraph-Extraction	Path-Retrieval	Implemented Instances
Structure-based Methods on Both Modules (Group (I): *SBE & SBR*)	SBE (-RW/KSE/PPR)	SBR (-EPR/SPR)	Our Instance: No.1
Semantic-Augmented Methods on Both Modules (Group (II): *SAE & I/OSAR*)	SAE (-EEMs/LLMs)	OSAR (-EEMs/LLMs)	Our Instances: No.2, 3, 4, 5
	SAE (-EEMs/LLMs)	ISAR (-EEMs/LLMs)	GCR (arXiv24) [72] Our Instances: No.6, 7, 8, 9
Semantic-Augmented Methods on Subgraph-Extraction (Group (III): *SAE & SBR*)	SAE (-EEMs/LLMs)	SBR (-EPR/SPR)	RoG (ICLR24) [71] GSR (EMNLP24) [45] Our Instances: No.10, 11
Semantic-Augmented Methods on Path-Retrieval (Group (IV): *SBE & I/OSAR*)	SBE (-RW/KSE/PPR)	OSAR (-EEMs/LLMs)	Our Instances: No.12, 13
	SBE (-RW/KSE/PPR)	ISAR (-EEMs/LLMs)	StructGPT (EMNLP23) [50] Our Instances: No.14, 15
Without Subgraph-Extraction Modules (Group (V): *SBR or I/OSAR*)	None	SBR (-EPR/SPR)	-
	None	OSAR (-EEMs/LLMs)	KELP (ACL24) [65]
	None	ISAR (-EEMs/LLMs)	ToG (ICLR24) [98], DoG (arXiv24) [74]

- Defined five types of GraphRAG instance groups.

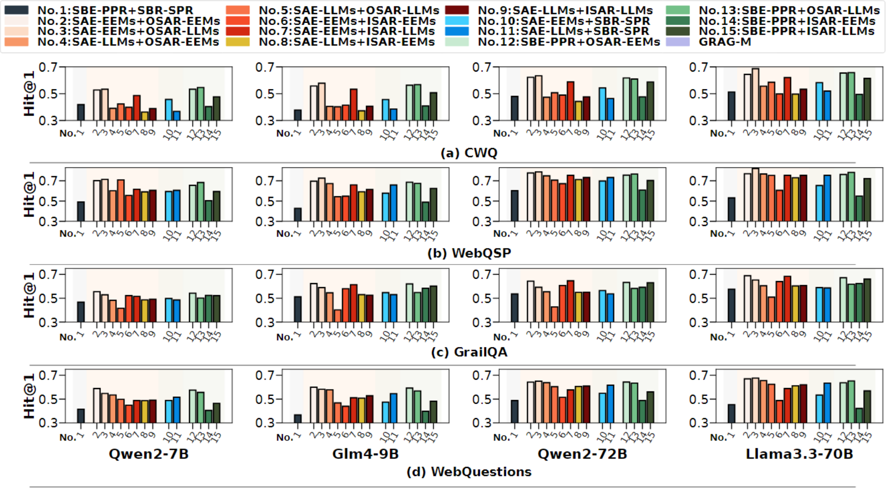
Existing Instances vs. LEGO-GraphRAG Instances

Table 3: Existing Instances vs. LEGO-GraphRAG Instances

GraphRAG Instances	WebQSP		CWQ	
	Hits@1	Recall	Hits@1	Recall
RoG [71] (RoG planning w/ChatGPT)	81.51	71.60	52.68	48.51
LEGO-RoG (RoG planning w/ChatGPT)	82.79	64.41	56.06	49.76
KELP [65] (one-hop w/gpt-4o-mini)	31.06	-	14.16	-
LEGO-KELP (one-hop w/gpt-4o-mini)	77.36	63.99	48.65	43.88
ToG [98] (w/Llama3-8B)	59.76	43.05	36.97	32.69
LEGO-ToG (w/Llama3-8B)	66.44	44.77	40.26	33.63

- exhibit performance comparable to their original counterparts.
- KELP implementation outperforms the original.

Reasoning Performance



Runtime of SE and PR

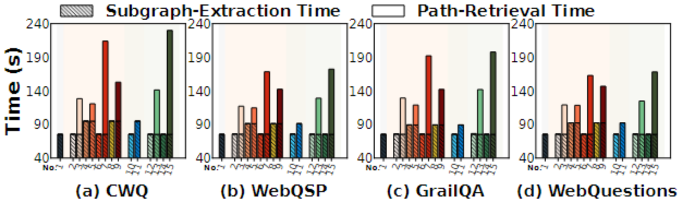


Figure 5: Runtime of SE and PR Modules for GraphRAG Instances

- Subgraphextraction is the primary bottleneck in GraphRAG runtime.
- ISAR-LLMs methods in the path-retrieval may result in unacceptably low query efficiency.

Token cost and Peek GPU memory

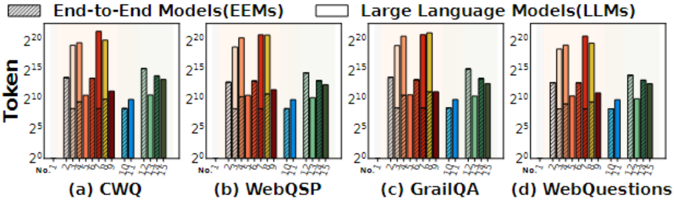
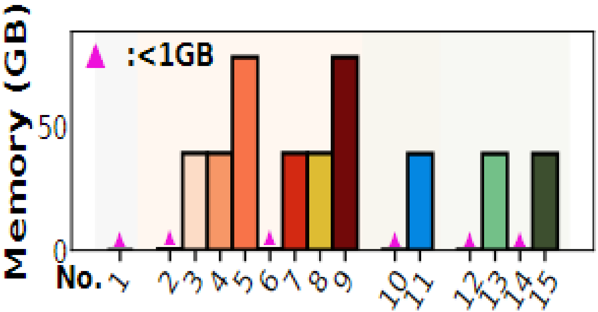
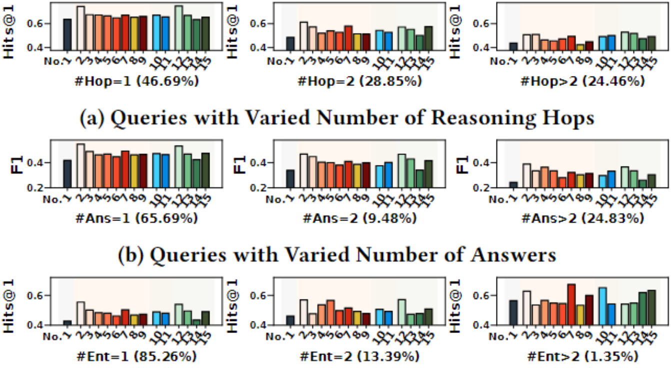


Figure 6: Token Costs for EEMs and LLMs in GraphRAG Instances





(c) Queries with Varied Number of Entities

Figure 7: Instance Performance w.r.t. Queries

Integrating Microsoft GraphRAG

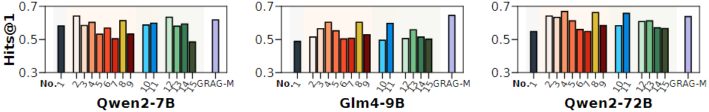


Figure 3: Results of LEGO-GraphRAG Instances and GraphRAG-M Instance on MetaQA Dataset

- implement an instance (GRAG-M): performing community detection on the graph; precomputing textual summaries for each community;

Efficiency of the Subgraph-Extraction

Table 6: Computational acceleration of PPR

Approximate PPR (Freebase)			Distributed PPR	
Method	**Recall**	**Ave. Time (s)**	**Method**	**Speedup**
RBS [107]	0.31	**0.51**	HGPA [31]	3.4–4.1×
Fora [110]	0.18	7.61	PAFO [109]	58.7×
TopPPR [113]	0.44	42.08	Delta-Push [39]	**123–162×**
Standard PPR	**0.96**	75.29	**Standard PPR**	1× (baseline)

Efficiency of the Subgraph-Extraction

Table 7: Performance and Cost Analysis of the SE Acceleration Methods on Freebase (About 100M Nodes, 300M Edges)

Method	Pre-time	Online-time	Recall	F1	WCC
Precomputation	19373s	19.02s	0.52	0.0021	**1**
Vector Database	14570s	**0.85s**	0.65	0.0023	12.86
Standard PPR	**0s**	75.29s	**0.96**	**0.0038**	**1**

Generation Quality of LLMs

Table 8: Performance of strategies designed to mitigate the limitations of LLM-generated outputs

Method	CWQ	WebQSP	GrailQA	WebQuestions
SBE+OSAR-LLMs	0.304	0.401	<u>0.401</u>	0.328
SBE+OSAR-LLMs (M-LLMs)	<u>0.349</u>	**0.438**	0.381	<u>0.348</u>
SBE+OSAR-LLMs (EEMs-S)	**0.381**	<u>0.427</u>	**0.476**	**0.353**