Stack2Graph: A Structured Knowledge Representation of Stack Overflow Data for Retrieval-based Question Answering

Technische Hochschule Augsburg

Lukas Kleybolte, Viviana Ventura, Alessandra Zarcone – Technical University of Applied Science Augsburg

Introduction

MOTIVATION

Answering programming questions requires aligning **natural language, code,** and **reasoning**

Stack Overflow is a rich resource, but existing datasets:

- treat posts as isolated pairs
- ignore structural relations

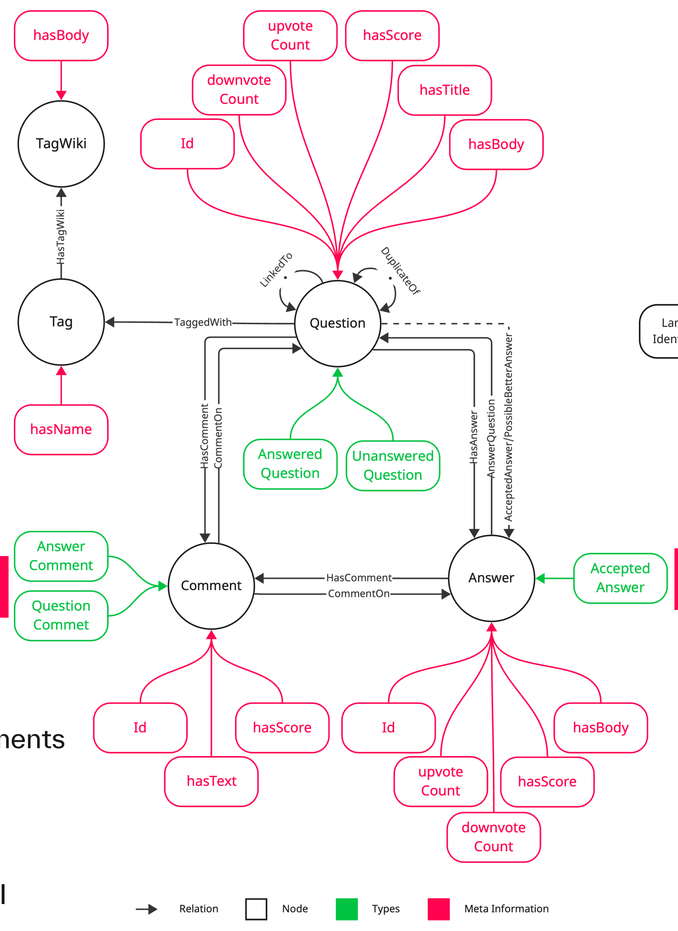
Our idea: Stack2Graph

We reconstruct Stack Overflow as a hybrid retrieval resource combining a large-scale RDF knowledge graph with language-specific vector databases

Goal:

Enable structured and multi-hop retrieval for LLM-based programming-based QA

KNOWLEDGE GRAPH SCHEMA



DESCRIPTIVE STATISTICS

Knowledge Graph

117M nodes | 275M edges

1.6B RDF statements

16M questions | 26.8M answers | 72.6M comments

Vector Database

25.6M indexed points

Dense vectors: 3584 dimensions

Top-10 language collections cover **82.7%** of all indexed points

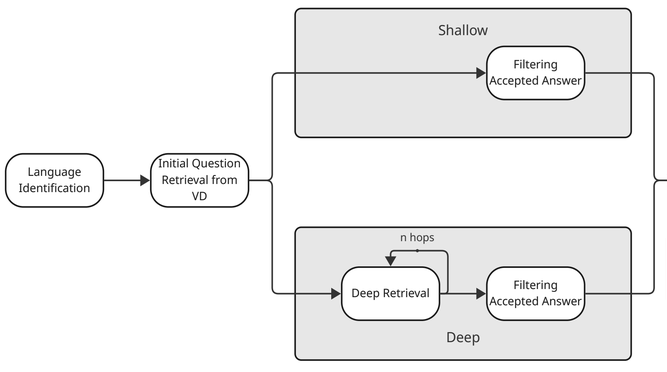
39 programming languages

Method

KNOWLEDGE GRAPH

Models Stack Overflow as **RDF triples**

Captures explicit relations between **questions, answers, comments, tags, duplicates, links, votes,** and **accepted answers**



VECTOR DATABASE

Stores **semantic representations** of Stack Overflow questions

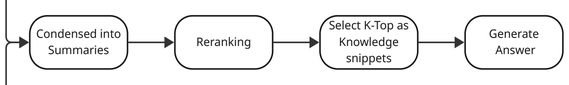
Sparse Embeddings: *BGE-M3*

Dense Embeddings: *Nomic-Embed-Code*

Chunked posts (**parent–child chunking**)

Connected with KG via ID

Organizes posts in **language-specific** Qdrant **collections**



RETRIEVAL CONFIGURATION

Zero-Shot: LLM answers directly without external knowledge

Shallow, Hop=0: Vector Database retrieval only; keeps questions with accepted answers

Deep, Hop=1: Vector Database retrieval + KG expansion via duplicate and linked relations

Optional summarization: applied to shallow and deep retrieval variants

Graph access: KG queried with **SPARQL**

MULTIPLE CHOICE QA PIPELINE

Identify programming language from the input question (regex/LLM)

Retrieve semantically similar questions using vector search in language-specific Vector Database collection (4 x top-k)

Expand candidates through the Knowledge Graph using explicit relations such as *DuplicateOf* and *LinkedTo*

Construct and summarize knowledge snippets from retrieved questions, answers, and comments

Rerank snippets with **bge-reranker-v2-m3**

Inject top-k-ranked snippets as context for the LLM

Result

CODEMMLU BENCHMARK

Hop=1 increases coverage by ~12-14 percentage points but also increases semantic drift.

Task	Hop=0	Hop=1	ZS Fallback
API	41.7%	53.8%	52.2%
DBMS	46.1%	59.9%	47.0%
Syntax	44.1%	57.5%	49.2%
Principles	25.9%	36.6%	68.8%

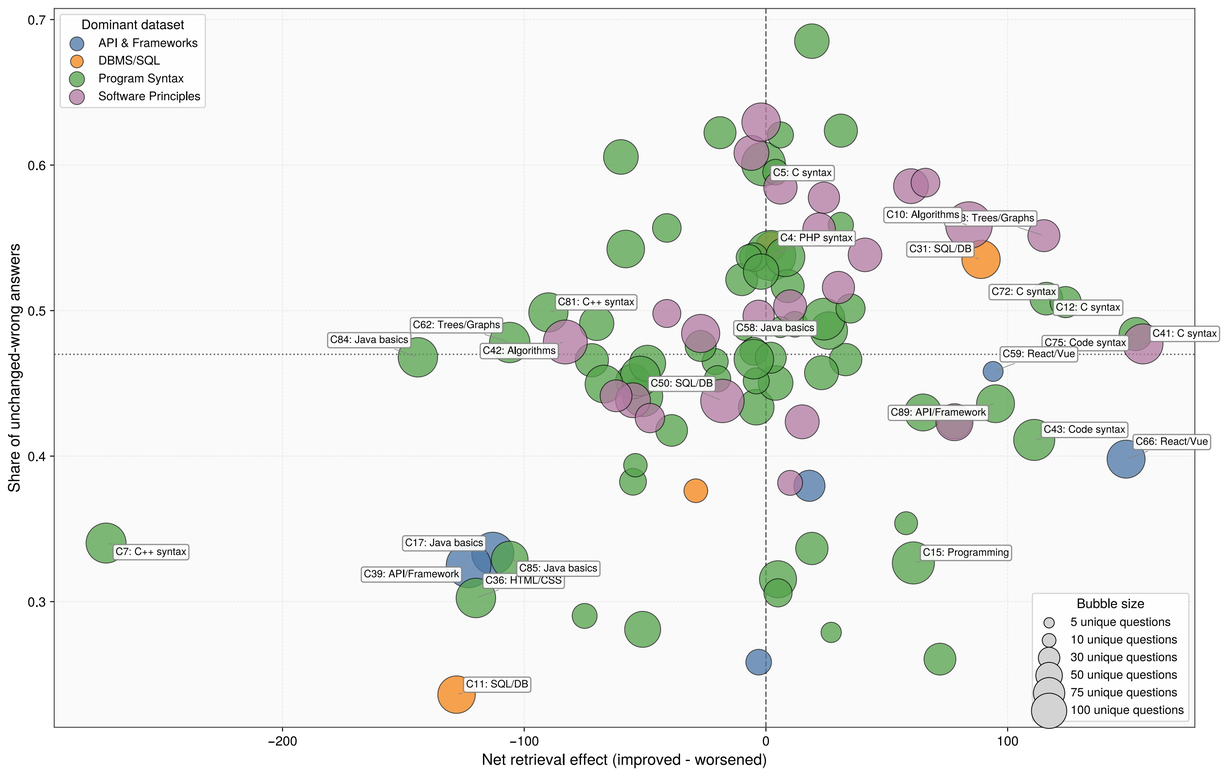
CLUSTER ANALYSIS

Questions embedded and grouped into semantic clusters using **PCA + MiniBatchKMeans**

Clusters reveal where retrieval helps or hurts across models and domains

Retrieval mainly benefits **API/framework** and **concept-oriented** questions

Syntax-heavy and SQL-related clusters are more prone to noisy or mismatched snippets



Conclusions

DISCUSSION

Retrieval helps when:

- Questions ask for API usage, framework conventions, or concepts
- Relevant SO evidence contains stable, community-validated knowledge
- Mid-sized general-purpose models lack enough parametric knowledge

Retrieval hurts when:

- Questions require exact syntax execution or formal reasoning
- Retrieved snippets are topically similar but semantically mismatched
- Hop=1 expansion introduces noisy linked threads

Even after hop-1 expansion, only 36.6-59.9% of questions receive at least one valid knowledge snippet

LIMITATIONS & FUTURE WORK

Static Hop=1 expansion can be noisy or "luck-based"
→ Dynamic, confidence-aware graph traversal

Static SO snapshot from June 2025
→ Time-aware retrieval for evolving APIs and frameworks

Pipeline-based hybrid retrieval
→ Learned integration of graph exploration and vector similarity

Retrieval limited to accepted answers
→ Credibility-, vote-, and time-aware evidence selection

Underused KG signals such as scores, comments, votes, and references
→ Score-aware ranking, relation weighting, and agent-based retrieval

Model	Software Principles			Programm Syntax			DBMS SQL			API & Framework		
	ZS	S2G	Δ	ZS	S2G	Δ	ZS	S2G	Δ	ZS	S2G	Δ
Llama3.2 1B	29.2	30.2†	+1.0	29.5	29.7†	+0.2	30.8	32.6†	+1.8	37.2	36.1†	-1.1
Llama3 8B	42.8	44.3*	+1.5	44.3	47.9*	+3.6	54.8	53.2†	-1.5	64.1	65.2*	+1.1
Qwen2.5 0.5B	32.3	31.9†	-0.4	32.5	31.4†	-1.1	36.0	34.2*	-1.8	40.2	38.1*	-2.1
Qwen2.5 1.5B	37.7	40.3*	**+2.7**	38.8	43.8*	**+5.0**	47.8	54.5†	**+6.7**	43.7	65.2*	**+21.5**
Qwen2.5 7B	63.1	62.9*	-0.1	61.2	60.1*	-1.1	66.3	67.6*	+1.3	83.7	82.0†	-1.7
Qwen2.5 Coder 0.5B	25.5	26.3†	+0.8	30.5	30.0†	-0.5	27.8	26.2†	-1.5	33.0	32.4*	-0.6
Qwen2.5 Coder 1.5B	41.8	40.6*	-1.2	45.2	43.3†	-1.9	41.4	44.2†	+2.8	53.9	56.3†	+2.4
Qwen2.5 Coder 7B	60.9	60.8†	-0.1	62.7	62.0†	-0.7	64.8	65.8†	+1.0	82.5	82.3†	-0.1
Qwen3 4B	62.9	62.4†	-0.5	64.0	63.6*	-0.3	66.8	69.7†	+2.8	80.5	79.9†	-0.6
Deepseek Coder 1.3B	23.4	25.1*	+1.6	27.4	27.4†	+0.1	27.5	28.3†	+0.8	27.0	27.2*	+0.3
Mistral 7B	34.5	36.3†	+1.9	39.9	43.2*	+3.3	43.4	48.8*	+5.4	47.2	60.6*	+13.4

† Shallow mode (hop=0) * Deep mode (hop=1)

Knowledge Graph

Vector Database

Code Repository

Contact: Lukas.Kleybolte@tha.de

KG & LLM: Knowledge Graphs and Large Language Models LREC 2026 Workshop