GRAPH-AWARE RETRIEVAL-AUGMENTED GENERATION: LEVERAGING KNOWLEDGE GRAPHS AND LLM-BASED EVALUATION

Âu Hồng Ân

[1] University of Information Technology - VNUHCM

What ?

We introduce an enhanced RAG framework with the following key contributions

- Overcomes naive RAG/LLM limitations in retrieval,generalization, and response quality.
- Replaces local optimization with entity-aware global analysis using knowledge graphs.
- Links responses to structured knowledge to minimize AI hallucinations.

Why ?

QA systems rely on knowledge retrieval and text generation, but naive RAG, LLMs often lack accuracy and reasoning capabilities.

Knowledge graphs boost reasoning via relational data and support efficient large-scale, real-time solutions.

Knowledge graphs boost reasoning via relational data and support efficient large-scale, real-time solutions.

Overview

RAG (Retrieval-Augmented Generation) is a method that combines information retrieval and language generation models to enhance question-answering or text-generation capabilities based on knowledge beyond the model's training data. However, basic approaches like naive RAG and large language models (LLMs) face challenges in information retrieval, generalization difficulties, and the reinforcement of repetitive responses. To address these limitations, we focus on a key improvement: RAG integrated with a knowledge graph. Incorporating a knowledge graph into RAG significantly enhances reasoning capabilities. Unlike naive RAG, which may get stuck in local optima, our approach uses [2] to pursue global optimization while explicitly modeling relationships between entities. This method also reduces hallucinations by tightly linking responses to the knowledge graph.

These advancements highlight the importance of integrating structured knowledge into the question-answering process. Future developments may focus on updating and expanding the knowledge graph, as well as optimizing inference speed for real-time applications. By combining these techniques, the question-answering solution can better meet demands across various domains

Description

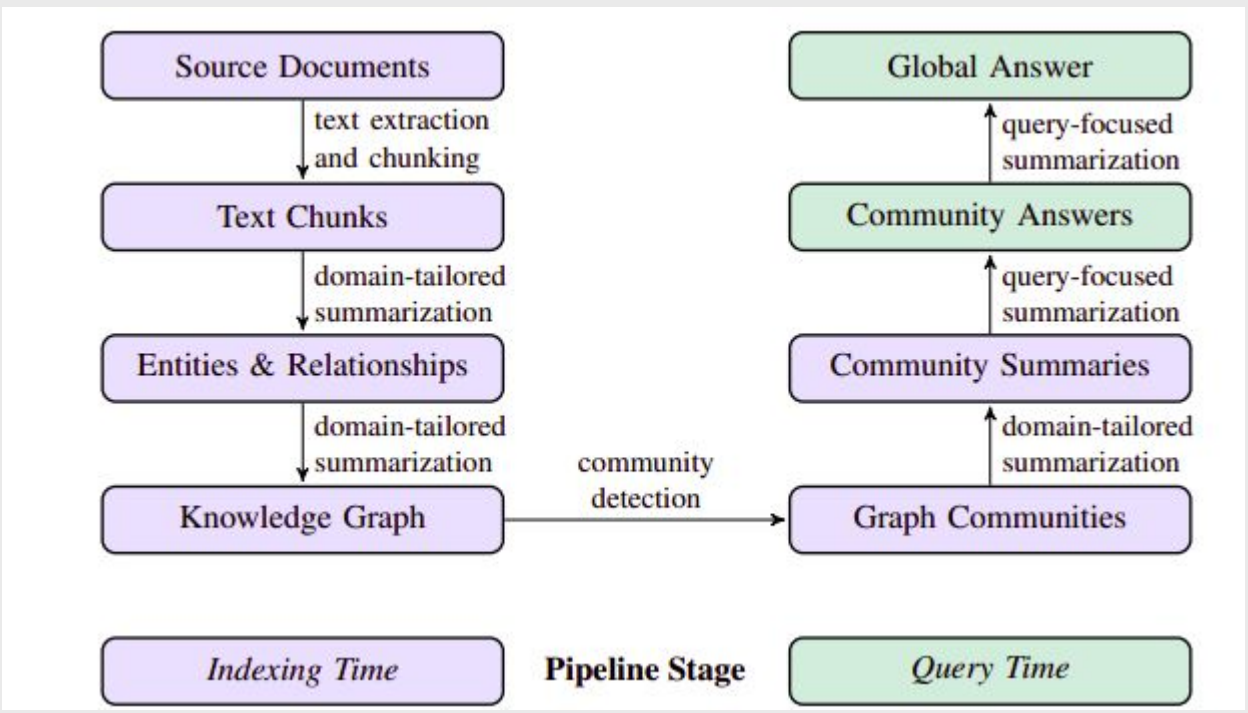

Figure 1: Graph RAG Pipeline with LLM-Derived Graph Index
The pipeline constructs a structured graph index from source documents, with nodes (entities), edges (relationships), and covariates (claims) extracted and summarized using domain-specific LLM prompts. Community detection (e.g., Leiden algorithm) partitions the graph for efficient parallel summarization, and the final answer is generated via query-focused aggregation of relevant community summaries.

NII

Âu Hồng Ân – Trường Đại học Công nghệ Thông tin - Đại học Quốc gia TP.HCM
TEL : ******* Email: anah.19@grad.uit.edu.vn**