

Revisión de la Literatura

Esta revisión de la literatura comenzará proporcionando una visión general de las tecnologías, marcos y metodologías de vanguardia utilizados en aplicaciones de RAG utilizando una base de conocimiento en grafo. Esto incluirá el preprocesamiento de documentos, la creación de grafos de conocimiento, la incrustación, la recuperación de información y la generación utilizando LLMs. Posteriormente, discutiremos algunos estudios relevantes sobre aplicaciones de RAG en campos no financieros, particularmente aplicaciones médicas, ya que pueden ser útiles al orientar el uso específico del dominio de la arquitectura RAG. Y después de eso, resumiremos algunos estudios sobre el uso de RAG, calidad de datos, etc., para aplicaciones FinTech, especialmente la planificación financiera, que será el enfoque de la propuesta de este documento.

Preprocesamiento y Almacenamiento de Datos

Las bases de datos de grafos han emergido como sistemas especializados diseñados para almacenar y gestionar datos en formatos de grafo, enfatizando las relaciones entre los puntos de datos.

Tecnologías como Neo4j, Amazon Neptune y ArangoDB ofrecen soluciones escalables para manejar estructuras de grafos complejas. Estas bases de datos permiten consultas y recorridos eficientes de las relaciones, lo que las hace ideales para aplicaciones que requieren una recuperación y manipulación de datos interconectados de alto rendimiento, lo cual es fundamental para los sistemas Graph RAG.

El almacenamiento y la recuperación efectivos en Graph RAGs dependen de estructuras de datos optimizadas y métodos de indexación que faciliten el acceso rápido a nodos y aristas relevantes. Se emplean técnicas como listas de adyacencia, indexación de caminos y particionamiento de grafos para gestionar grandes grafos de manera eficiente. Lenguajes de consulta como Cypher y Gremlin permiten capacidades de consulta expresivas, lo que permite a los sistemas RAG recuperar subgrafos o relaciones pertinentes necesarias para la generación de contenido informada.

Basándose en estas tecnologías fundamentales, **Peng et al. (2024)** discutieron la construcción y la indexación de bases de datos de grafos en el contexto de GraphRAG, enfatizando tanto las bases de datos de grafos de código abierto como las autoconstruidas. Exploraron varios métodos de indexación, incluyendo la indexación de grafos, texto y vectores, para mejorar la eficiencia de la recuperación [1]. Su trabajo destaca la importancia de estrategias de indexación personalizadas para mejorar el rendimiento de los sistemas RAG.

Un grafo de conocimiento es una representación estructurada de entidades del mundo real y sus interrelaciones, organizada en un formato de grafo donde los nodos representan entidades y las aristas representan relaciones. Construir un grafo de conocimiento implica crear una representación estructurada de la información donde las entidades (como personas, lugares o conceptos) son nodos, y las relaciones entre ellas son aristas que conectan estos nodos. El proceso comienza con la recopilación de datos de diversas fuentes, incluyendo bases de datos estructuradas, documentos de texto no estructurados y contenido web. Estos datos diversos se preprocesan para asegurar la consistencia y la calidad, lo que implica la limpieza para eliminar inexactitudes y la normalización para estandarizar formatos [2].

El núcleo de la construcción de un grafo de conocimiento implica extraer entidades y las relaciones entre ellas utilizando técnicas de procesamiento de lenguaje natural como el reconocimiento de entidades nombradas y la extracción de relaciones. Se desarrolla una ontología o esquema para definir las categorías de entidades y las relaciones permisibles, proporcionando una estructura formal que sustenta el grafo de conocimiento. Esta ontología sirve como un plano que guía la integración de datos y asegura la consistencia semántica a lo largo del grafo [3].

Bibliography

- [1] BOCI PENG *et al.*, “Graph Retrieval-Augmented Generation: A Survey,” vol. 37, no. 4.
- [2] Lisa Ehrlinger and Wolfram Wöß, “Towards a Definition of Knowledge Graphs.”
- [3] Paulheim H., “Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods.”
- [4] Xiaoxin He *et al.*, “G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering.”
- [5] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao, “GRAG: Graph Retrieval-Augmented Generation.”
- [6] Zetian Hu, Weijian Xie, Xuefeng Liang, Yuhui Liu, Kaihua Ni, and Hong Cheng, “WeKnow-RAG: An Adaptive Approach for Retrieval-Augmented Generation Integrating Web Search and Knowledge Graphs.”
- [7] Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald, “REANO: Optimising Retrieval-Augmented Reader Models through Knowledge Graph Generation,” vol. 1.
- [8] Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald, “TRACE the Evidence: Constructing Knowledge-Grounded Reasoning Chains for Retrieval-Augmented Generation.”
- [9] Cui Long, Yongbin Liu, Chunping Ouyang, and Ying Yu, “Bailicai: A Domain-Optimized Retrieval-Augmented Generation Framework for Medical Applications.”
- [10] Chao Jin *et al.*, “RAGCache: Efficient Knowledge Caching for Retrieval-Augmented Generation.”
- [11] Wenyu Huang, Mirella Lapata, Pavlos Vougiouklis, Nikos Papasarakantopoulos, and Jeff Z. Pan, “Retrieval Augmented Generation with Rich Answer Encoding.”
- [12] Rong-Ching Chang and Jiawei Zhang, “CommunityKG-RAG: Leveraging Community Structures in Knowledge Graphs for Advanced Retrieval-Augmented Generation in Fact-Checking.”
- [13] Junde Wu, Jiayuan Zhu, and Yunli Qi, “Medical Graph RAG: Towards Safe Medical Large Language Model via Graph Retrieval-Augmented Generation.”
- [14] I. de Zarzà, J. de Curtò, Gemma Roig, and Carlos T. Calafate, “Optimized Financial Planning: Integrating Individual and Cooperative Budgeting Models with LLM Recommendations.”
- [15] Komal Azram Raja, “Domain Specific Data Quality Framework.”
- [16] Thomas Sepanosian, Zoran Milosevic, and Andrew Blair, “Scaling AI adoption in finance: modellingframework and implementation study.”