

GraphRAG-Enabled AI Agents: A Framework for Enterprise Knowledge Base Building

Loys Belleguie¹ 早矢仕 晃章² 眞田 貴央¹

Teruaki Hayashi Takahiro Sanada

1. Introduction

Despite the expectation that integrating data from diverse fields will drive innovation, enterprises continue to face substantial technical challenges arising from fragmented, heterogeneous, and semantically complex datasets. These issues limit comprehensive visibility and hinder effective governance within the enterprise data ecosystems. To overcome these challenges, we propose a novel graph retrieval augmented generation (GraphRAG) framework that combines large language model-based AI agents with enterprise-specific Knowledge Graphs. Our method significantly outperforms traditional RAG techniques in natural language querying, automatic classification, and data discovery tasks, laying the groundwork for advanced semantic engineering in enterprise data management.

2. Issues and Our Research Questions

Large enterprises with a global business scope must manage multiple data sources constantly produced by interconnected business entities. The main entry point for the observability of these corporate data landscapes is provided by the metadata extracted from traditional databases such as relational database management systems (RDMS), business applications such as Enterprise Resource Planning (ERP), Supply Chain Management (SCM), and Data Warehouses (DWH), where data may be combined for use in business analytics and AI models. Data catalogs, namely management tools for those metadata, have become a central tool for IT engineers as well as for end users, such as business analysts, data scientists, or AI models consuming those datasets.

Technical or operational metadata, that is, table names, columns, data types, or schemas, are usually easily imported into data catalogs because they are part of the data structure themselves; however, business (i.e., descriptive) metadata are often found to be incomplete by data stewards (metadata managers). This is the first hurdle in the quest for high-quality metadata that can provide relevant information regarding the underlying data.

The second major obstacle is the gap between the available descriptive metadata and (1) their interpretation within the business context and (2) the scarcity of semantic relationships with other datasets. The former can be addressed either manually by Subject-Matter Experts (SME) or, to a certain extent, by using Generative AI tools. The latter, on the other hand, calls for a broader analysis of the whole metadata landscape. The endeavor to consolidate this information in a common metadata database is

often limited by the disconnection between dataset domains (silos). This results in a lack of holistic visibility of the entire corporate data landscape, thereby significantly reducing the ability to leverage existing data for business analysis.

This research presents a novel methodology to address two specific tasks: the lack of full *intelligibility* of the underlying data thread and a methodology to build a fully *observable* and *searchable* corporate data knowledge base.

Then, we conduct specific experiments based on real use-case hypotheses to validate our methodology.

RQ1: How is the searchability, intelligibility, and governance of the corporate data knowledge base improved with the GraphRAG agentic framework?

RQ2: How much does the extended GraphRAG knowledge base improve the agentic AI framework, especially the sub-LLM agents' response, compared with the standard vector embedding setup?

3. Related Works

3.1 Corporate Data Knowledge based on Metadata

The development of a robust corporate data knowledge base is fundamentally reliant on the effective management and strategic utilization of metadata, hence the importance of data catalogs [1] [2] [3]. Metadata are, in many cases, the only piece of information that data users can access before starting any data related activity. Metadata provides essential context, detailing the lineage, structure, semantics, quality, and access policies of data assets. The focus has shifted towards enhancing these catalogs with more dynamic and intelligent metadata capabilities, leveraging RAG/LLMs to automate the creation of rich textual descriptions for data catalog assets.

3.2 AI Framework based on sub-LLM agents

In the context of the agentic framework, the concept of a *Mixture of Experts* (MoE) has been introduced to address the increasing demand for systems capable of handling tasks with escalating complexity, which necessitates a shift from general-purpose capabilities to nuanced, domain-specific performance. Although this concept has been known for decades, it has become mainstream research following the AI boom and foundational work by Google Research [4]. The concept has been extended to the macro-level framework of a *Multi-Agent System* (MAS), where each agent acts as a dedicated expert [5]. The sub-LLM expression has also recently been used to designate (often local) models trained on a specific subject. In our study, on enterprise context we

¹ Quollio Technologies

² School of Engineering, the University of Tokyo

use the local implementation of the LLM model that is trained and run in a protected (secured) environment, together with generic open-access agents. These models can be fed sensitive data, and their attached agents are designed with guardrail features to avoid unintended use.

3.3 Knowledge Graph powered LLM in datasets query tasks

The concept of using a structured knowledge base, in this instance with a graph database, to improve the RAG of LLMs compared with standard vector embedding methods has gained considerable momentum since early 2024 [6], [7] and the introduction of the GraphRAG implementation by Microsoft Research [8].

The main difference between the direct vector embedding of contextual documents and GraphRAG is that GraphRAG creates a structured view of the content by extracting logical elements (semantic entities, paragraphs, etc.) into a knowledge graph. Then, it provides a unified knowledge base for metadata, business glossaries, and semi-structured documents. The KG can be queried to extract precise contextual elements to feed the RAG process. It is noteworthy that vector embedding can still be used in parallel. For the standard RAG, embeddings are calculated on specific text elements (attributes of KG nodes) and are clearly identified and accessible from a KG query. For example, the semantic information of a database table, which is a node in our KG with attributes such as physical and logical names, description, and list of columns. These attributes are used to calculate the embedding vectors and stored as additional attributes of the node. The same applies to chunks of parsed documents, where the chunk itself is used for embedding, but the chunk is not a random number of characters; it is a logical part of the document (e.g., a paragraph). Unlike traditional vector searches, which retrieve isolated text chunks, the graph enriches LLM inputs with interconnected AI agents built with the capacity to search for information outside their pre-trained scope. This concept is comparable to the Model Context Protocol (MCP) recently proposed by Anthropic¹, where many different information sources can be added to feed the LLM. While searching for additional information directly on the Internet is fine for general topics, it should be carefully monitored when working with corporate data, that is, data domains where confidentiality may be mandatory or at least controllable.

4. Proposed Method

To address dataset discovery (searchability), intelligibility, and governance tasks in the context of enterprise-specific data management, we propose a framework based on a combination of GraphRAG powering LLM agents inspired by MoE designs. The framework consists of two main parts: (1) the knowledge base (built on KG), and (2) the agentic AI system. A conceptual and functional diagram is shown in Figure 1. It is worth noting that this framework is a “KG-enhanced LLM” but we also use LLM to build the KG through the document parsing and entity extraction, it is therefore also a “LLM-enhanced KG” [6].

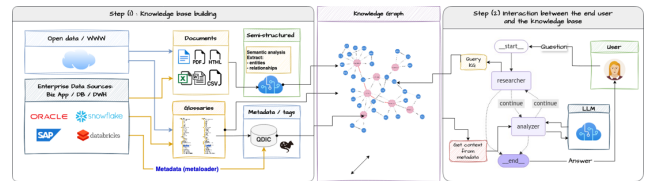


Figure 1 KG-AI Agent Framework

4.1 Knowledge graph

Our KG is used as a customized and fully controlled knowledge base that can be *interrogated*, that is, queried in the traditional SQL sense, but also searched with keywords or natural language questions, or explored like a network. We used the Neo4j graph database.

We built a knowledge base from 3 main sources of data:

1. The core metadata lake from the Quollio metadata management engine (QDIC),
2. The company internal documents (a priori confidential), with specific term glossaries, and
3. Open-source documents and business glossaries, etc.

The important point of this process is that all data, such as pure metadata and glossaries provided as CSV files, HTML pages, or PDF files, are parsed and loaded into a common graph database space to create the main knowledge graph. The documents are analyzed, and entities are extracted to build additional semantic glossaries. Once the KG is built, we run several procedures and algorithms to enrich the knowledge base with new relationships based on several similarity metrics (described below), attach glossary terms automatically, or create domains. This process is semi-automatic in the sense that some monitoring (i.e., a *human-in-the-loop*) remains necessary to avoid unavoidable inaccuracies, even with current state-of-the-art LLM models.

A KG is an appropriate tool for building a knowledge base. It is a collection of triples $\mathcal{G} = \{(\mathcal{E}_i, \mathcal{R}_k, \mathcal{E}_j) \subseteq \{\mathcal{E}\} \otimes \{\mathcal{R}\} \otimes \{\mathcal{E}\}\}$ in which an entity \mathcal{E}_i (node) is in relationship with another entity \mathcal{E}_j , through a link \mathcal{R}_k . Each entity in the set $\{\mathcal{E}\}$ contains as many attributes as necessary $\mathcal{E}_i(e_n, e_m, \dots, e_l)$, likewise the relationships $\mathcal{R}_k(r_n, r_m, \dots, r_l)$. The problem is then to (1) build the relevant sets $\{\mathcal{E}\}$ and attributes, and (2) calculate or extract the correct sets $\{\mathcal{R}\}$. All relevant information is contained in these triples, and additional insights can be discovered through lexical or topological graph exploration.

In the context of metadata and parsed documents, the attributes e_n can be text with semantic or simple technical information (e.g., data type), but they can also carry vector embeddings $e_b \in \mathbb{R}^d$ calculated from any combination of these textual contents. The latter is used to compute vector-based similarities between entities: for any entity $\mathcal{E}_i(\vec{e}_b)$, we consider the set $\{\mathcal{E}_j(\vec{e}_c) | \cos(\vec{e}_b, \vec{e}_c) > \alpha\}$, where α is the similarity-score threshold.

¹ <https://www.anthropic.com/news/model-context-protocol>

4.2 Agentic AI workflow

The core of the system is the agentic framework, which acts as the interface between the knowledge base and end user. The details of the implementation are developed in the next section, but in summary, we create a series of agents equipped with specific tools in two broad categories: (1) KG queries and search and (2) LLM contextual analyses. Figure 2 illustrates the overall architecture.

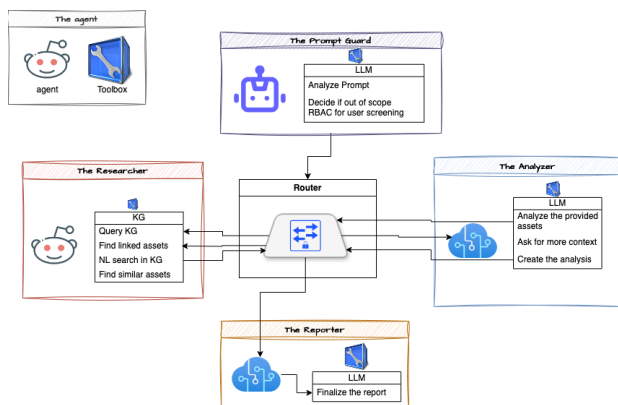


Figure 2 Agentic Workflow Concept

This framework is an MAS, which is by itself a major research topic in the Collaborative AI field. It can be represented as $\mathcal{A} = \{a_i\}_{i=1}^n$, where a_i is defined with a series of features, a model (architecture) m including contextual or constraints parameters, a set of objectives \mathcal{O} , and an input x and an output y , resulting in $a_i = \{m, \mathcal{O}, x, y\}$. In our case, agent models fall into two main categories.

- (1) The KG query: x is the request to find a certain type of information in the KG. A realistic question could be: “Find all metadata connected through lineage downstream from Table A up to 5 hops with all tags attached to each node along the path,” or “Find if there are any datasets that may be linked with regulation X.” The input x is a business question whereas the output y is a structured dataset that will be used by the other agents.
- (2) The reasoning agents that call GraphRAG-LLMs¹: x is the KG-enhanced prompt and y gives a response that may require more context, i.e., send a request back to (1), continue in the reasoning process or finalize the answer to the initial user question.

4.3 Sub-LLM

In this implementation, we only build two main agent types: (1) the generic (open access) agent (e.g., using OpenAI API) and (2) local LLM models running in a dedicated and secured, environment (e.g., Ollama¹). The former is used to handle all non-sensitive documents, build glossaries from open data, etc., while the latter is used for private enterprise data. As a direct extension of KG-enhanced LLM, it is possible to perform fine-tuning to

build local LLM models using the context extracted from the knowledge base.

5. Experimental Setup

5.1 Knowledge base building

The core metadata are a direct mapping of the QDIC “metadata database” into a graph database. This is a straightforward process, which is not detailed here. We then added all the available glossaries. The most important and difficult part is the inclusion of semi-structured data, such as web pages (HTML) or documents in PDF, which are parsed and analyzed, bringing invaluable contextual information to the knowledge base. In this study, we used only two types of formats, HTML pages and PDF, for documents that we wanted to be part of our local KG, in contrast with the information that the LLM agent would access *on the fly* from the Internet. The former would be, for example, company confidential information or open-source data that is deemed particularly important for the business use case, from which we may want to extract specific entities and their definitions.

The parsing process decomposes documents into sections and chunks, following their original structure, and then loads them as nodes in the graph. In addition, we extract the entities and store them in a business glossary. We use LlamaParse² for this study. Our choice was motivated by the quality, limited free access to the API, and ease of use. Entity extraction is performed with an LLM that we provide, and we tested several solutions and retained AzureOpenAI for the sake of simplicity and quality of the generated responses.

5.2 Agents design

LangGraph is used as the basis for the proposed framework. The agents are based on the prebuilt *ReAct* type, as described in [9].

Each agent acts as an independent entity using predefined tools that will take the output from the previous agent, perform an action, use the *chosen* tool, and feed the result of this tool to the next agent. It is worth noting that all reasoning occurs at this stage; the agent decides which tool or LLM to call. We defined three main categories of agents in our experiments.

1. A guardrail selects only those requests that are in scope.
2. A search agent capable of querying a KG to fetch certain categories of metadata assets, paths (e.g., lineage), entities, and their related document text chunks directly into a graph. It can also run natural language searches within the KG.
3. Then, an analyzer focusing on sub-LLM calls is based on context.

5.3 Application in business context

We applied our framework to an illustrative yet realistic use case. Large enterprises, especially those with a global business scope, often face a common issue: “how can we guarantee that our products are, or will still be in the future, compliant with all regulations around the world, EU, US, or Asian markets?” This is particularly relevant for many Japanese companies, whose exports

¹ <https://ollama.com/>

² <https://www.llamaindex.ai/llamaparse>

constitute a large part of their business. A concrete case is for example a simple announcement from the EU official web site stating that European Parliament voted to approve a proposal aimed at simplifying the Carbon Border Adjustment Mechanism (CBAM) that will introduce a threshold that would exempt companies importing low-value consignments from the reporting obligations¹. This simple announcement already contains several keywords that only experts or a fine-tuned AI model will detect as major potential disruption factors for exports to the European market: CBAM, EU, threshold, reporting obligations, and so on. These terms are related to the carbon footprint of products, which may or may not comply with the EU import regulations.

A deep analysis of the corporate situation with regards to this use case, often boils down to the question from the business analysts: “Explore in the corporate datasets, evaluate any potential risks related with this announcement, provide the data entry points with explanation.” We designed a list of use cases, summarized in Table 1, for each set of questions/expected answers to test our setup.

Table 1 Business use case examples

Business question	Scenario / Data connections / Value
1. Regulatory Compliance Tracking (REACH, CBAM)	EU announces changes to chemical regulations or carbon border adjustments / Product BOM → Chemical components → Supplier data → Regulatory terms / Automatically identify which products/suppliers are affected by regulatory changes
2. Supply Chain Risk Assessment	Geopolitical event threatens semiconductor supply from specific regions / Supplier location data → Product BOM → Manufacturing schedules → Financial impact / Proactive identification of at-risk products and alternative suppliers
3. ESG / Sustainability Reporting	New sustainability reporting requirements for product carbon footprint / Manufacturing data → Energy consumption → Supplier emissions → Product lifecycle / Automated discovery of all data points needed for compliance reporting
4. Cross-Domain Impact Analysis	Raw material price fluctuation or shortage announcement / Commodity prices → BOM costs → Product margins → Sales forecasts / Rapid assessment of business impact across departments

To answer these questions, we used the following experimental setup:

¹ <https://www.europarl.europa.eu/news/en/press-room/20250515IPR28461/parliament-supports-proposals-to-simplify-eu-carbon-leakage-instrument>

- (1) Ingesting in the KG, all relevant information on the use case topic and create the business term (entity) glossary
- (2) Building similarity relationships between the corporate metadata and the documents from (1)
- (3) Running the Agentic Framework with the business questions above.

By scanning internal documents about the suppliers (e.g., supplier audit of their carbon emissions), matching with the relevant metadata (product’s definition and Bill Of Materials location) and regulations terms (e.g., “EU Digital Product Passport”), KG query agents will provide the relevant information to the analyzer agent as well as all the nodes connected to the supplier’s material flows, compliance history, and sustainability reports. This structured context reduces LLM hallucinations and allows the analyzer to prioritize high-risk suppliers for manual audits or flagging discrepancies between reported PCF values and raw material sourcing data.

We focused on the first business case, Regulatory Compliance Tracking, and the list of questions in Table 2.

Table 2 List of questions for business case (1) in Table 1

1	“How can we guarantee that our products are, or will still be in the future, compliant with all regulations around the world, EU, US Asia markets?”
2	“We received an alert that substance ‘Bisphenol A’ (CAS: 80-05-7) has been added to the SVHC candidate list. Which of our products might be affected?”
3	“Which suppliers have outdated or missing REACH compliance certificates for materials used in our EU-export products?”
4	“The EU announced updates to REACH Annex XVII adding restrictions on PFAS chemicals. How can we identify affected products in our catalog?”

6. Results and Discussion

We consider the first use case: The goal is to generate a compliance readiness assessment for products tagged in the EU market. Therefore, we will ask the agents to provide recommendations about which data sources should be checked for REACH compliance. In this case, we expect the agents to follow the following workflow.

- (1) Search for tables related with REACH regulation: KG similarity search query: Agent chooses the method
- (2) Find linked tables (e.g. by lineage) to extend the context: KG query
- (3) Find related regulatory information: KG similarity search query → return to Analyzer agent
- (4) The analyzer agent returns answers or asks for more context.

Two main processes are involved in this workflow. The first targets the KG to search for the relevant assets, as shown in Figure 3, which shows the result from the Search agent call to the KG

with the question “Find all linked assets by lineage from table COMPLIANCE_RISK_FACT.”

Additional information is then provided by the search agent to obtain regulatory documents related to the questions (similarity search).

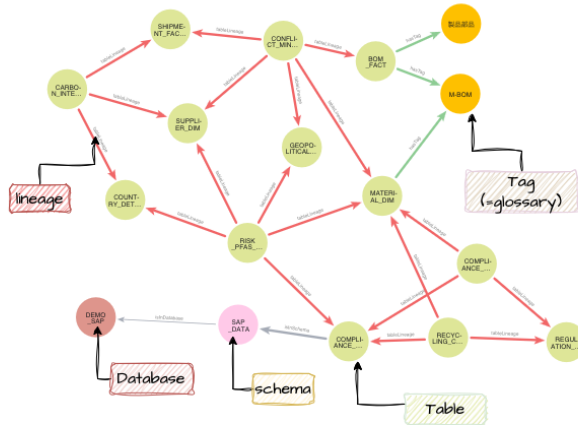


Figure 3 Linked assets returned by the search agent

Table 3 presents the results obtained by the search agent. Note that these “entities” are only tokens extracted from regulatory documents used in the initial stage (ingestion of documents in the KG) and are not related to a specific taxonomy. They all point to document sections that can be used by the LLM as context as well as by the business analyst.

Table 3 Entities returned by the search agent

Entities	Similarity score
Compliance Checks	0.51
Substance Identification	0.48
Communication in the Supply Chain	0.48
Assessment of Regulatory Needs	0.47
Extended Safety Data Sheets	0.47

The second category is the analysis per se (“Analyzer agent”), which uses all information to provide the final response in Table 4. As expected, different prompts provide different responses, which provides a good illustration of what can be achieved.

Table 4 Output from the analyzer agent

Query: Find tables related to product compliance and show their relationships with regulatory data. Focus on tables that might contain information relevant for EU regulations.
Conclusion: The identified tables, 'COMPLIANCE_RISK_FACT' and 'REGULATION_DETAILS', are crucial for understanding product compliance in relation to EU regulations. Key findings include: 1. Core Compliance Architecture: COMPLIANCE_RISK_FACT table with REGULATION_ID and RISK_SCORE columns, connected to SUPPLIER_DIM and MATERIAL_DIM via foreign keys, provides

infrastructure for risk-based assessment required by REACH and upcoming EU regulations.

2. Regulatory Readiness Indicators: (a) CARBON_INTENSIVE_ANALYSIS table with REGULATION_NAME column aligns with CBAM requirements for carbon content tracking; (b) Extracted critical materials (Lithium, Cobalt, REEs) from EU documents map to materials likely in MATERIAL_DIM, supporting Critical Raw Materials Act compliance; (c) M-BOM tags on battery-related tables support EU Battery Regulation 2023/1542 requirements.

3. Multi-Language Compliance: Japanese metadata (製品部品) alongside English equivalents enables global regulatory coordination, essential for multinational REACH compliance and safety data sheet requirements.

4. Supply Chain Visibility: The 5-hop graph traversal capability and foreign key relationships {Supplier → Material → Product} provide technical infrastructure for EU Due Diligence Directive compliance, enabling end-to-end traceability.

5. Semantic Validation: The 0.44 similarity score between COMPLIANCE_RISK_FACT and REACH "Compliance Checks" proves the framework bridges technical and regulatory domains. While functional, scores above 0.7 would provide higher confidence for substance identification critical to REACH's "no data, no market" principle.

The 8.5-second response time enables real-time compliance queries during product development. This infrastructure can answer complex questions like "Which products contain Annex XVII substances from high-risk suppliers?"—essential for EU market access but impossible with traditional data catalogs. The discovered data structures align with major EU regulatory requirements, validating GraphRAG as a practical solution for regulatory intelligence.

Confidence: 0.92

7. Conclusion

This study successfully demonstrates that the GraphRAG framework, combining Knowledge Graphs with LLM-based agents, provides a powerful solution for enterprise data discovery and business intelligence goals. Our experimental results validate the approach and answer both research questions, while revealing significant advantages over traditional RAG implementations.

7.1 Research questions addressed

RQ1: Enhanced Searchability, Intelligibility, and Governance

Searchability: The use of KG enhances the granularity and precision of the RAG. In addition, it allows the search for assets through semantic matching via natural language queries while monitoring similarity scores. Multi-path discovery through lineage, semantic similarity, and tag propagation (e.g., M-BOM tags) enables comprehensive data discovery beyond simple table-name matching.

Intelligibility: The framework bridges the semantic gap between metadata and business concepts: the initial business query points to a set of tables that demonstrates how the original data catalog structure becomes understandable in the context of business analysis. Multi-language support (e.g., 製品部品 tags) further enhances global intelligibility.

In addition, multi-agent reasoning provides focused expertise, which is impossible with single-model approaches. Each agent contributes to specific capabilities, such as query understanding, graph exploration, relationship discovery, and analytical synthesis.

Governance: The complete workflow {Prompt parsing → Asset search → Discover linked assets → Analyzer} is fully traceable from the KG queries to the agent inputs/outputs, documenting the reasoning path from query to discovery. This provides explainable results that are essential for the verification of regulatory compliance.

RQ2: GraphRAG Superiority over Standard Vector Embedding
Our framework significantly outperforms the standard vector embedding RAG in several key respects.

Multimodal Discovery: While vector RAG finds only semantic similarities, GraphRAG combines semantic matching, structural relationships (lineage, foreign key), and metadata properties (tags, columns) for comprehensive discovery.

This framework combines the semantic understanding of LLMs with the structural precision of graph databases. This dual capability enables queries like “Which products from high-risk suppliers contain REACH-restricted substances?”—requiring both semantic interpretation {"high-risk", "restricted"} and structural navigation {Supplier → Material → Product → Regulation}.

Precision with Explainability: Semantic similarity scores provide measurable confidence levels, unlike opaque vector similarities. The graph structure explains why connections exist that are critical for regulatory compliance.

Performance at Scale: KG allows the selection of only the relevant information from millions of data points (nodes) to feed the LLM RAG. The same process using a standard RAG would require scanning the entire catalog and PDF documents. A KG is also fully scalable and versatile; it can dynamically incorporate any kind of assets, metadata, glossaries (taxonomy or ontologies), and parsed documents. These assets are *interconnected* through lexical (e.g., Jaccard), vector embedding similarity, or more sophisticated graph algorithm clustering methods (e.g., community detection, such as the Louvain method).

7.2 Final remarks

This study confirms that the GraphRAG concept is a practical solution for enterprise data challenges, particularly in use cases where semantic understanding of the entire data landscape and hidden relationships across fragmented systems determine success or failure.

A core novelty lies in the integration of metadata and documentation into a common *interconnected* graph, and leveraging the application of graph algorithms combined with LLM. The integration of LLMs is realized in several steps: (1) at the initial conversion of semi-structured data into a knowledge graph after parsing in logical chunks, entity extraction, and embedding calculation, (2) discovery of hidden relationships based on different business cases, and (3) interaction between the end user and the knowledge base in a natural language.

Our initial experiments using a *regulatory compliance* business case validated this framework. This suggests applicability to other enterprise challenges requiring relationship-aware intelligence, such as supply chain optimization, risk management, ESG reporting, and cross-domain impact analysis.

References

- [1] D. Haynes, Metadata for information management and retrieval, Second edition ed., London: Facet Publishing, 2018.
- [2] O. V. G. Olesen-Bagneux, The enterprise data catalog, First edition ed., Beijing: O'Reilly, 2023.
- [3] P. Strengholt, Data management at scale, Second edition ed., Beijing: O'Reilly, 2023.
- [4] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton and J. Dean, Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer, arXiv, 2017.
- [5] K.-T. Tran, D. Dao, M.-D. Nguyen, Q.-V. Pham, B. O'Sullivan and H. D. Nguyen, "Multi-Agent Collaboration Mechanisms: A Survey of LLMs," 10 January 2025.
- [6] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang and X. Wu, "Unifying Large Language Models and Knowledge Graphs: A Roadmap," IEEE Transactions on Knowledge and Data Engineering, vol. 36, p. 3580–3599, July 2024.
- [7] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang and S. Tang, "Graph Retrieval-Augmented Generation: A Survey," 15 August 2024.
- [8] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness and J. Larson, From Local to Global: A Graph RAG Approach to Query-Focused Summarization, arXiv, 2024.
- [9] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan and Y. Cao, "ReAct: Synergizing Reasoning and Acting in Language Models," 6 October 2022.