Neo4j-Powered Graph-RAG System for Financial Insights on the Colombo Stock Exchange

Bashitha Shamila
Department of Computer Science & Engineering,
University of Moratuwa,
Sri Lanka
bashitha.22@cse.mrt.ac.lk

Shaveen Silva
Department of Computer Science & Engineering,
University of Moratuwa,
Sri Lanka
shaveen.22@cse.mrt.ac.lk

Samadhi Talagala
Department of Computer Science & Engineering,
University of Moratuwa,
Sri Lanka
samadhi.22@cse.mrt.ac.lk

Keywords—Graph RAG, Knowledge Graphs, Financial Analysis, Large Language Models (LLMs), Information Retrieval

I. INTRODUCTION

Financial annual reports contain rich but unstructured corporate information, making it difficult to efficiently extract relationships such as directors, subsidiaries, auditors, and ownership structures. Knowledge graphs provide a structured way to model these relationships and integrate heterogeneous information sources [1]. With recent advances in retrieval-augmented generation (RAG), graph-based retrieval can be combined with large language models to improve factual accuracy and context grounding in downstream analysis [4]. In this work, we transform Colombo Stock Exchange (CSE) annual reports into a Neo4j-based financial knowledge graph and integrate it with an LLM-driven Graph-RAG pipeline that supports natural-language financial querying.

Our method introduces a focused retrieval strategy that extracts entities only from the most relevant text segments, overcoming LLM context limitations. This targeted approach enables near-complete entity capture and more accurate graph construction than full-document extraction.

II. LITERATURE REVIEW

Knowledge graphs provide a structured way to model entities and relationships from unstructured data, enabling cleaner integration and reasoning across heterogeneous sources [1]. Prior work highlights the importance of refinement techniques for improving entity consistency [2] and shows how financial knowledge graphs can capture corporate structures such as ownership links and board interlocks [3]. Recent advances in Graph-RAG demonstrate that combining subgraph retrieval with large language models improves factual grounding and complex reasoning in question answering tasks [4], [5], supporting our hybrid Neo4j-based approach.

III. MATERIALS AND METHODS

Our system follows a streamlined end-to-end pipeline that converts CSE annual reports into a Neo4j financial knowledge graph and supports natural-language financial querying.

A. Document Processing

Annual report PDFs are segmented using a context-aware semantic chunking strategy with a fixed window of approximately 1000 characters, preserving paragraph and section boundaries. Each chunk is vectorized using the Google Text Embedding 004 model. The total input budget for the embedding model was set at 3000 tokens per document. The resulting vectors are stored in a vector index for efficient retrieval in the subsequent stage.

B. Targeted Entity Extraction

Gemini 2.5 Pro then uses RAG to retrieve relevant chunks, extracting key financial entities (directors, auditors, etc.) into JSON. This targeted approach minimizes errors and is limited by a 500-token extraction context window. The success of this extraction strategy is measured by the Data Capture metric, defined as the percentage of ground-truth entities and relationships successfully extracted from the document set. Specifically, a successful data capture is defined by the extraction of (1) the correct number of entities and (2) all relevant relationships associated with specific high-value fields (e.g., entity type: 'DIRECTOR', relationship type: 'HOLDS_POSITION') from the subset of documents used for benchmarking.

C. Entity Consolidation & Graph Construction

Entity names were unified using a custom fuzzy matching pipeline integrated with a Human-in-the-Loop (HIL) clustering interface to ensure high fidelity. The process employed type-specific similarity functions (person: strict surname & initials; company: normalized token-sort & word-overlap). HIL review allowed experts to manually merge/split clusters before variants were mapped to their unique canonical entity nodes in Neo4j, enforcing a rigid

corporate-relationship schema (e.g., DIRECTS, OWNS). (Fig. 1).

D. Natural-Language Querying

User questions are handled through a LangGraph/ LangChain agent that maps natural-language intent to Cypher. Complex questions are decomposed into modular sub-queries; such as entity lookup and relationship traversal, for interpretable retrieval.

E. Self-Correction & Validation

The agent applies lightweight self-verification loops that check intermediate results. If a sub-query returns incomplete or invalid output, the query is reformulated and retried to ensure stable and accurate responses.

IV. Results and Discussion

The final knowledge graph contains over 6,000 unique entities including companies, directors, executives, subsidiaries, auditors, products, and sectors comprehensively capturing the corporate structure of the Colombo Stock Exchange (CSE). Our vectorization-based retrieval approach demonstrates substantial improvements over baseline full-document processing. The initial approach, which fed entire reports directly to LLMs, resulted in approximately 50% data loss due to context window limitations. In contrast, our question-driven retrieval pipeline vectorizing reports with Google Text Embedding 004 in ChromaDB and retrieving the top-5 relevant chunks per query achieved over 85% entity capture on a manually annotated subset, representing a reduction in data loss.

To evaluate question-answering performance, we manually extracted ground-truth answers for a benchmark of 10 queries covering simple lookups, relational queries, and multi-hop reasoning. Each query was executed using (1) our Graph RAG pipeline with Neo4j subagent and (2) a standard RAG baseline without graph reasoning. Answers were scored on a 1–10 correctness scale. The Graph RAG method achieved an average accuracy of 9.2/10, compared with 6.1/10 for standard RAG, a 50% relative improvement, with the largest gains in multi-hop relational queries. A sample query, *"Who are the directors common to both Company A and Company B in 2023?"*, demonstrates this advantage. Our system generated correct Cypher with query decomposition, retrieved the intersecting director set, and matched ground truth perfectly. The RAG baseline returned incomplete lists and hallucinated names due to semantic confusion across chunks. The Neo4j subagent's query decomposition and self-correction retry mechanism reduced query failure rate from ~30% to <2%, significantly improving reliability.

Beyond question-answering, the structured graph enables advanced analytics: detecting board interlocks, tracing ownership chains, quantifying auditor concentration, and analyzing sector diversification tasks infeasible with

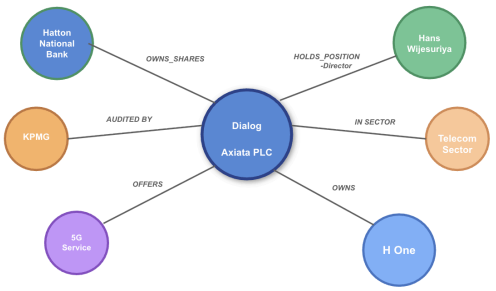

Fig. 1. Entities and relationships associated with a given company in the knowledge graph

vector-only RAG systems, demonstrating Graph RAG's unique value for financial intelligence applications.

The full implementation, including the data processing pipeline and graph construction scripts, is available in our GitHub repository.

V. Conclusion

This work demonstrates an end-to-end Graph-RAG pipeline that converts unstructured annual reports into structured, analysis-ready financial insights. The use of context-aware extraction and dual-step entity matching reduces data loss and improves consolidation accuracy. Although the current approach does not yet handle tabular financial data, the underlying pipeline is generalizable and can be applied to other domains and document collections beyond the CSE with minimal adaptation.

VI. Future Work

Future work includes improving extraction of tabular financial data using layout-aware parsing, reducing latency in natural-language-to-Cypher generation through caching and query optimization, and adding real-time analytics with precomputed graph metrics to support faster financial insights.

References

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 2, pp. 494–514, Feb. 2022, doi: https://doi.org/10.1109/TNNLS.2021.3070843.

[2] H. Paulheim, "Knowledge Graph refinement: a Survey of Approaches and Evaluation Methods," Semantic Web, vol. 8, no. 3, pp. 489–508, Dec. 2016, doi: https://doi.org/10.3233/sw-160218.

[3] A. Arun, F. Dimino, T. P. Agarwal, B. Sarmah, and S. Pasquali, "FinReflectKG: Agentic Construction and Evaluation of Financial Knowledge Graphs," arXiv.org, 2025, doi: https://doi.org/10.1145/3768292.3770363..

[4] B. Peng et al., "Graph Retrieval-Augmented Generation: A Survey," arXiv.org, 2024. https://arxiv.org/abs/2408.08921

[5] C. Ma, Y. Chen, T. Wu, A. Khan, and H. Wang, "Large Language Models Meet Knowledge Graphs for Question Answering: Synthesis and Opportunities," arXiv.org, 2025. https://arxiv.org/abs/2505.20099

[6] The Neo4j Graph Data Science Library Manual v1.8 - Neo4j Graph Data Science," Neo4j Graph Database Platform. https://neo4j.com/docs/graph-data-science/current/