LLM-RAG for Financial Question Answering: A Case Study from SET50

Naphatta Chinaksorn* and Dittaya Wanvarie†
Department of Mathematics and Computer Science
Faculty of Science, Chulalongkorn University
Bangkok, Thailand, 10330
Email: *6434441923@student.chula.ac.th, †dittaya.w@chula.ac.th

2025 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) | 979-8-3315-0702-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICAIIC64266.2025.10920730

***Abstract*—This study compares the performance of a traditional relational database with a financial knowledge graph in retrieval-augmented generation (RAG) settings. The knowledge graph contains stock closing prices and financial statement data from companies in the SET50 index, focusing on key financial metrics and market performance indicators. The study examines two main aspects: response time and the accuracy of large language models (LLMs) in query generation. To evaluate the efficiency of both databases, complex queries—such as cross-industry financial ratio comparisons and trend analysis over time—are utilized. The experimental results indicate that the graph database requires less time to retrieve results. Furthermore, the LLM can translate natural language into queries for both the graph and relational databases with a similar level of accuracy.**

I. Introduction

In recent years, the growing complexity of investment decisions has driven the need for advanced tools that can handle sophisticated financial queries. Investors often seek to compare financial ratios across companies within the same industry or analyze trends over different periods. Traditional relational databases such as MySQL, which utilize a tabular structure, often struggle with these complex queries due to the need for extensive joins across multiple tables, resulting in inefficiencies and slower response times. Recent studies compared the performance of querying relational databases with NoSQL databases, especially graph databases [1]–[5]. The studies found that graph databases, such as Neo4j, outperformed relational databases, such as Oracle, MySQL, MariaDB, and PostgreSQL when the queries require complex joins between many tables.

With their graph-based structure, knowledge graphs offer a more efficient alternative for managing multidimensional, interconnected data. Knowledge graphs can significantly reduce the complexity of querying interconnected data points by directly linking nodes and edges, allowing faster insight and decision-making [6]–[8].

The construction of financial knowledge graphs has shown significant promise in enhancing data integration and query performance across financial datasets [9]. One of the challenges of querying a knowledge graph lies in the query languages, such as SPARQL and Cypher, which interact with the database and typically require writing expertise. This complexity restricts the ability of the general audience to take advantage of graph databases. However, recent advancements in large language models (LLMs) have made it possible to translate human intent or questions expressed in natural language into code [10]–[12]. This integration has been applied to chatbot-based query systems, where LLMs help translate natural language questions into database queries, allowing for intuitive user interactions. Combining KGs and LLMs in financial applications provides a robust framework for managing complex queries and delivering insights more efficiently.

This research aims to assess the performance of knowledge graphs in Retrieval-Augmented Generation (RAG) settings. We will compare the accuracy of a large language model (LLM) in translating natural language text into both SQL and Cypher queries. Additionally, we will measure the elapsed time of an end-to-end chatbot system to evaluate the usability of financial graph RAG in addressing investors' inquiries.

The structure of this paper is organized as follows: In §II, we discuss relevant research. Next, in §III, we outline our database construction, query design, and evaluation metrics. We present the experimental results and discuss notable findings in §IV and §V, respectively. Finally, in §VI, we summarize our contributions and suggest directions for future research.

II. Related Research

In the financial domain, a knowledge graph can be constructed from both unstructured text—such as financial research papers and reports—and structured data, like tabular financial statements and daily market prices [9], [13]–[15]. Two widely used data models for knowledge graphs are the RDF model, which uses predicates to represent relationships, and the property graph model, such as Neo4j, which allows nodes to possess an arbitrary number of properties and relationships among them [16]. Additionally, Zehra et al. [6] highlighted the effectiveness of knowledge graphs in improving query performance and accuracy, showcasing their potential in financial reporting systems where quick retrieval of relevant data is critical.

The performance characteristics of graph and relational databases have been thoroughly studied to assess their suitability for various types of queries. Sholichah et al. [17] highlighted that while relational databases excel with simple queries, graph databases are superior in handling complex, multi-dimensional queries, such as those requiring real-time

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:46 UTC from IEEE Xplore. Restrictions apply.

analysis or cross-dimensional comparisons. Do et al. [3] conducted a comparison of four query types on Neo4j and MySQL databases and found that querying the graph database is generally faster than the relational database, especially for tasks involving multiple joins, recursion, aggregation, and pattern matching. However, Kotiranta et al. [4] provided contrasting results, indicating that although a graph database may surpass a relational database in join scenarios, this is not necessarily the case for aggregation queries. One possible explanation for this difference is that the data size tested in Kotiranta et al.'s study was significantly larger than that used in the study by Do et al. [3].

Large Language Models (LLMs) are increasingly being used as tools for financial question-and-answer systems. However, the training data for these LLMs often lacks adequate financial information, which can hinder their ability to address questions effectively in the financial domain. One potential solution is to fine-tune the LLMs using relevant financial data [18], [19]. Another approach is Retrieval-Augmented Generation (RAG), which retrieves related documents and enables the LLM to generate answers based on that information [20], [21]. Rather than merely retrieving candidate answers, RAG can also integrate additional knowledge to enhance the generation of responses [22]. This knowledge can come in various forms, including unstructured text and knowledge graphs [23], [24].

Integrating Large Language Models (LLMs) with knowledge graphs has shown promise in enhancing natural language processing (NLP) capabilities in financial applications. Pan et al. [25] conducted a survey on the utilization of LLMs and knowledge graphs. Knowledge graphs can improve both the training and application of LLMs. Conversely, LLMs can also be utilized to construct knowledge graphs. Furthermore, when LLMs are used as reasoning agents on knowledge graphs, they are closely related to Retrieval-Augmented Generation (RAG) techniques. Since LLMs are also employed to generate code in various programming languages [10], [26], they can also be used to create query languages for databases [11], [12]. Utilizing LLMs can enhance user experience with knowledge graphs by converting natural language queries into database queries.

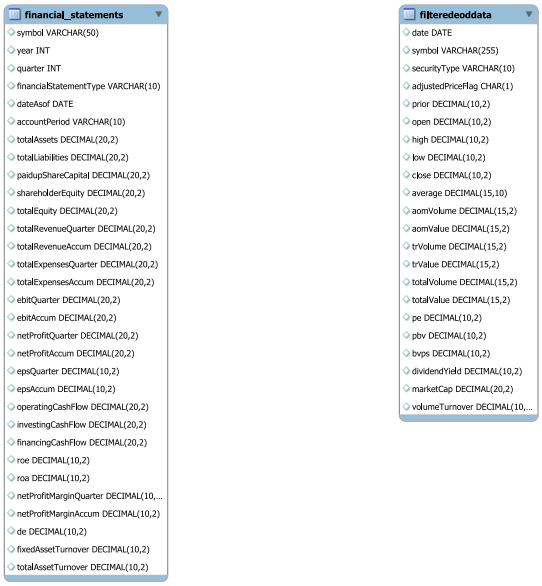

Fig. 1. MySQL database schema for SET50 financial data.

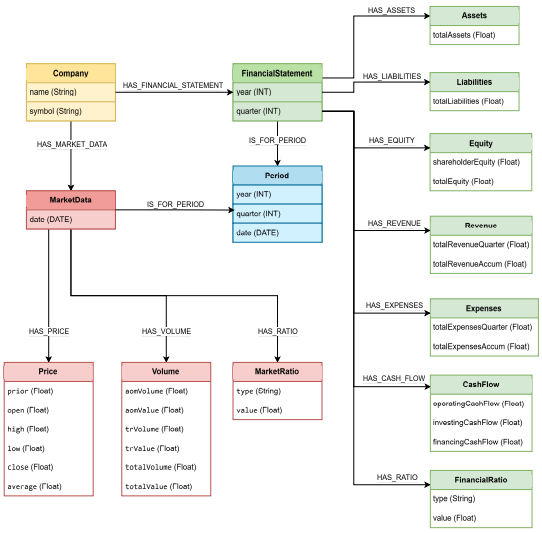

Fig. 2. Neo4j Knowledge Graph schema for SET50 financial data.

III. Methodology

A. Database Construction

We obtained data from SET's API via SET Smart Marketplace. The dataset includes daily end-of-day data and quarterly financial statements data from Year 2019-2023. Both data are in JSON format.

Two database systems were constructed to store and query the financial data of SET50 companies: a traditional MySQL relational database and a Neo4j Knowledge Graph. Each system was designed to handle the same dataset, enabling performance comparisons in terms of query speed and accuracy.

The MySQL database is composed of two tables: the financial statement table and the end-of-day table. Notably, neither table has a primary key. The database schema is illustrated in Fig. 1. The financial statement table contains 882 rows, while the end-of-day table has 5,850 rows.

In contrast, we have designed the graph database in Neo4j, adapting from the schema described in [9]. This schema includes 14 types of nodes (entities), 13 types of relationships, and 32 property keys. The end-of-day market data is divided into four nodes: date, price, volume, and market ratio. The financial statement data is organized into seven nodes: assets, liabilities, equity, revenue, expenses, cash flow, and financial ratios. Additionally, there is a period table that indicates the quarter of the year. The schema is depicted in Fig. 2. In total, the graph database contains 58,702 nodes and 59,394 relationships.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:46 UTC from IEEE Xplore. Restrictions apply.

Listing 1
EXAMPLE OF HUMAN-WRITTEN QUERIES

```
# Original question, type 1
What was BJC's total revenue in Q1 2019?

# SQL query
SELECT totalRevenueQuarter FROM financial_statements WHERE symbol = 'BJC' AND year = 2019 AND quarter = 1 LIMIT 1;

# Cyper query
MATCH (c:Company {symbol: 'BJC'})-[:HAS_FINANCIAL_STATEMENT]->(fs:FinancialStatement {year: '2019', quarter: '1'})
MATCH (fs)-[:HAS_REVENUE]->(r:Revenue)
RETURN r.totalRevenueQuarter AS TotalRevenue

################################################################################

# Original question type 2
What was PTTEP's Price-to-Earnings (P/E) ratio on September 1, 2023?

# SQL query
SELECT pe FROM filteredEODData WHERE symbol = 'PTTEP' AND date = '2023-09-01' LIMIT 1;

# Cypher query
MATCH (mr:MarketRatio{symbol: "PTTEP", type: "PE", date: "2023-09-01"})
RETURN mr.value AS PERatio

################################################################################

# Original question type 3
Compare ADVANC's Return on Equity (ROE) in 2019 with 2022.

# SQL query
SELECT year,Quarter, roe FROM financial_statements WHERE symbol = 'ADVANC' AND year IN (2019, 2022);

# Cypher query
MATCH (fs_2019:FinancialStatement {year: '2019', symbol: 'ADVANC'}) -[:HAS_RATIO]
    ->(roe_2019:FinancialRatio {type: 'ROE'}),
    (fs_2022:FinancialStatement {year: '2022', symbol: 'ADVANC'}) -[:HAS_RATIO]-> (roe_2022:FinancialRatio {type: 'ROE'})
WHERE fs_2019.quarter = fs_2022.quarter
RETURN DISTINCT fs_2019.quarter AS Quarter, roe_2019.value AS ROE_2019, roe_2022.value AS ROE_2022 ORDER BY Quarter

################################################################################

# Original question type 4
How did BDMS's ROE affect its profitability?

# SQL query
SELECT year,Quarter, roe FROM financial_statements WHERE symbol = 'BDMS' AND year BETWEEN 2019 AND 2021;

# Cypher query
MATCH (fs:FinancialStatement {symbol: 'BDMS'}) -[:HAS_RATIO] ->(roe{type: 'ROE'})
WHERE fs.year = "2019" OR fs.year = "2020" OR fs.year = "2021"
RETURN fs.year AS Year, fs.quarter AS Quarter, roe.value AS ROE ORDER BY Year, Quarter
```

B. Query Design

We developed four types of financial questions manually:

1) Financial statement figures
2) Market prices and information
3) Comparisons
4) Analysis questions

We then translated these questions into SQL and Cypher queries. Examples of the questions and their corresponding queries can be found in Listing 1. The query for analysis questions will search the database to help answer the question. However, the answer will be generated by the LLM.

A chatbot interface was created using Streamlit and the llama3−70b−8192 model to translate user inputs into database queries.

C. Performance Evaluation

We evaluate the process from two perspectives: accuracy and elapsed time. Specifically, we report two accuracy measurements in the following areas:

1) Query generation by the LLM
2) Response generation by the LLM

Human judges evaluated the accuracy of the query results and the semantic alignment of the responses with user intent, without focusing on the specific query structure. Query generation was evaluated on the basis of time and accuracy, with correctness determined by whether the generated query produced the expected result, regardless of its structure.

We applied BLEU, SacreBLEU, and BERTScore to evaluate chatbot-generated responses, focusing on both syntactic and semantic aspects across languages and database systems. BLEU and SacreBLEU measure syntactic accuracy by comparing the phrasing and structure of generated responses with expected answers. BERTScore, in contrast, evaluates semantic similarity by assessing how well the generated response captures the intended meaning, which is crucial for complex

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:46 UTC from IEEE Xplore. Restrictions apply.

TABLE I
DATABASE FETCH USING GENERATED QUERIES

Database	Language	Time (s)
MySQL	Thai	0.0024
	English	0.0015
Neo4j	Thai	0.0051
	English	0.0251

TABLE II
QUERY GENERATION

Database	Language	Time (s)	Accuracy (%)
MySQL	Thai	36.98	83
	English	26.17	80
Neo4j	Thai	23.70	88
	English	21.98	88

financial queries where meaning takes precedence over exact wording.

We measured the time required for query generation (LLM processing), data retrieval (database execution), and response generation (LLM output). By focusing on these metrics, we provided a comprehensive assessment of the chatbot's performance in generating accurate and linguistically aligned responses.

IV. RESULTS

A. Direct Query Performance

The direct query performance test measured the data retrieval speed using manually written SQL for MySQL and Cypher for Neo4j, averaging query execution times over ten iterations. MySQL demonstrated faster query response times for this dataset (0.001489 seconds vs. Neo4j's 0.009887 seconds). However, this result reflects the relatively small dataset used. As datasets grow larger and queries become more complex, Neo4j's performance may surpass MySQL due to its ability to efficiently handle relationships and interconnected data.

The time it takes to fetch data from the database is presented in Table I. The data indicate that MySQL outperforms Neo4j in translating Thai and English questions.

B. LLM Performance

The performance of chatbot-assisted queries was evaluated by examining response times and semantic accuracy for financial queries in Thai and English.

We evaluated the translation time from question to query and the accuracy of the LLM, as summarized in Table II. Neo4j outperformed MySQL in translation times for both Thai and English, with shorter generation times observed across both languages.In terms of accuracy, the LLM achieved higher success rates when generating Cypher queries for Neo4j (88%) compared to SQL queries for MySQL (83% in Thai and 80% in English). This can be attributed to the LLM's compatibility with Neo4j's graph-based syntax, which better aligns with the structured relationships in the dataset. Interestingly, the accuracy remained consistent across languages. Overall, Neo4j demonstrated superior performance in accuracy compared to MySQL.

TABLE III
RESPONSE GENERATION USING GENERATED QUERIES

Database	Language	Time (s)	Accuracy (%)
MySQL	Thai	52.45	70
	English	52.11	63
Neo4j	Thai	38.36	75
	English	44.72	68

We utilize the generated queries to retrieve knowledge from the database, allowing the LLM to create the response. The results are displayed in Table III. Compared to query translation, generating natural language responses from MySQL takes significantly more time than from Neo4j. Furthermore, the natural language responses generated in Thai exhibit higher accuracy compared to those produced in English.

Table IV shows that Thai responses consistently achieved higher BLEU and SacreBLEU values than English responses, reflecting the chatbot's stronger performance in Thai. However, English responses achieved higher BERTScore value, indicating better semantic alignment in English. This may be attributed to the chatbot's familiarity with structured and formalized financial expressions commonly found in Thai datasets.

TABLE IV
EVALUATION METRICS FOR GENERATED RESPONSES

Database	Language	BLEU	SacreBLEU	BERTScore
MySQL	Thai	0.3745	39.5054	0.8269
	English	0.0682	12.1697	0.8779
Neo4j	Thai	0.3650	39.4870	0.8460
	English	0.0840	14.1990	0.8780

In terms of syntactic alignment, there was little difference between the two databases for Thai responses, with MySQL scoring 0.3745 in BLEU and 39.5054 in SacreBLEU, compared to Neo4j's 0.3650 and 39.4870. However, Neo4j performed better for English responses with a SacreBLEU score of 14.1990, surpassing MySQL's 12.1697. For semantic preservation, there was little difference between the two databases for both Thai and English responses. MySQL had a BERTScore of 0.8269 for Thai and 0.8779 for English, while Neo4j scored 0.8460 for Thai and 0.8780 for English.

These findings underscore the trade-offs between syntactic fidelity and semantic accuracy. MySQL appears well-suited for straightforward, syntax-focused tasks, while Neo4j excels in scenarios requiring deeper semantic understanding, especially in multilingual and context-rich environments.

V. DISUCSSION

While we expected Neo4j's relationship model to outperform MySQL in retrieving results, the MySQL database's simplicity, with only two tables and no complex joins, gave it an edge in this scenario. In addition, the database was constructed using data from SET50 companies, which resulted in a relatively small data set. The faster query time

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:46 UTC from IEEE Xplore. Restrictions apply.

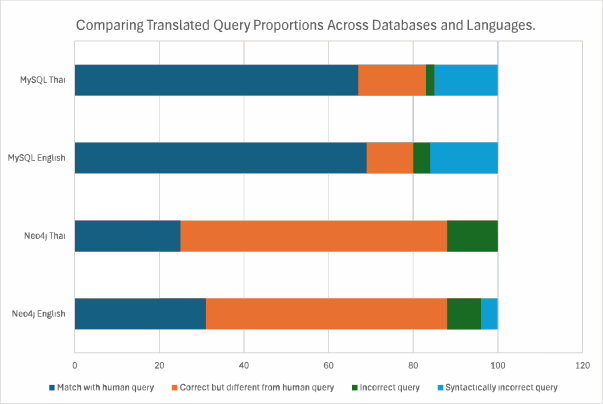

Fig. 3. Comparing Translated Query Proportions Across Databases and Languages.

of MySQL (0.001489 seconds) reflects the small data set and the simple schema, while the query time of Neo4j of 0.009887 seconds, although slower, may perform better with larger, interconnected data sets. However, as the size of the database scales, Neo4j's performance is expected to improve relative to MySQL due to differences in algorithmic complexity. MySQL's query performance typically scales with $O(n)$, where n is the total number of rows in the database. In contrast, Neo4j scales with $O(n)$, where n is the size of the relevant labels, which remains smaller than the total number of rows. This suggests that Neo4j may exhibit better efficiency in handling large, interconnected datasets.

The time required to translate text to query format for English queries is generally faster than for Thai queries. This difference is due to the fact that LLMs are more effective in English, a widely supported language, than in less-resourced languages like Thai. However, by fine-tuning the LLM, we can reduce the time it takes to generate queries. Another possible explanation for the delay is that the LLM might rephrase the question in English, which requires additional processing time.

In Table II, we evaluate the accuracy of the queries based on the results obtained. Furthermore, we analyze the queries generated from LLM in Fig. 3 to determine whether they differ from human-written queries. Although the generated SQL queries generally align well with the human-written ones, the generated Cypher queries rarely match. Despite this, the results remain accurate. This is because the LLM prioritizes the logic of the query to ensure that it retrieves the intended data, even if the structure of the query varies. However, the LLM may sometimes select unnecessary columns, change column names, or employ different query constructs to achieve the same result. We provide an example of an LLM-generated Cypher query in Listing 2, including the Response Answer and the Expected Answer for comparison. The LLM-generated Cypher query retrieves the same result as the human-written query, highlighting its ability to interpret user intent accurately. However, the LLM version may include minor syntactic differences, reflecting its flexibility in constructing queries. In contrast, the human-written query is typically more concise and standardized, designed for optimal readability and performance.

Regarding the Response Answer and Expected Answer, although their phrasing differs slightly, the semantic content is identical. This demonstrates the LLM's ability to generate accurate and semantically aligned responses, even when the exact phrasing does not match.

Overall, both the query and response comparison showcase the LLM's effectiveness in ensuring logical correctness and semantic alignment, making it a valuable tool for financial question answering.

VI. Conclusion and Future Work

This study evaluated the performance of MySQL and Neo4j in a financial question-answering system integrated with LLM. Both databases demonstrated satisfactory query generation accuracy (80–88%), with similar performance in many areas. Neo4j, however, was better suited for handling complex, interconnected financial data, thanks to its graph-based architecture, which enabled superior semantic alignment and efficient execution of multi-dimensional analyses, such as trend evaluations and cross-company comparisons. MySQL excelled in faster response times for simpler queries but faced challenges with more intricate tasks due to its reliance on complex joins and tabular structure. These findings underscore Neo4j's scalability and its capability to manage complex relationships, making it the better choice for advanced financial applications requiring semantic richness and data interconnectivity.

Future work will focus on expanding the knowledge graph by incorporating a wider range of financial data and improving the database structure to better accommodate the types of questions investors frequently ask. This will involve gathering additional real-world questions from investors to refine the data model and ensure that the database is tailored to the most relevant financial queries. Additionally, efforts will be made to enhance the chatbot's performance, including improving query generation accuracy and minimizing mistakes in selecting the correct data values. Addressing these areas will enhance the efficiency and accuracy of the system, making it more practical for financial decision-making.

Acknowledgment

The authors would like to acknowledge the use of Grammarly for assistance in improving the language quality of this paper.

References

[1] W. Khan, E. ahmed, and W. Shahzad, "Predictive performance comparison analysis of relational nosql graph databases," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 5, 2017.

[2] R. Čerešňák and M. Kvet, "Comparison of query performance in relational a non-relation databases," *Transportation Research Procedia*, vol. 40, pp. 170–177, 2019, tRANSCOM 2019 13th International Scientific Conference on Sustainable, Modern and Safe Transport.

[3] T.-T.-T. Do, T.-B. Mai-Hoang, V.-Q. Nguyen, and Q.-T. Huynh, "Query-based performance comparison of graphnbsp;database and relational database," in *Proceedings of the 11th International Symposium on Information and Communication Technology*, ser. SoICT '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 375–381.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:46 UTC from IEEE Xplore. Restrictions apply.

Listing 2
COMPARISON OF LLM-GENERATED CYPHER QUERY, RESPONSE ANSWER, AND EXPECTED ANSWER FOR PTTEP'S P/E RATIO ON SEPTEMBER 1, 2023

```
# LLM-generated Cypher query
MATCH (mr:MarketRatio{symbol: 'PTTEP',type: "PE",date: "2023-09-01"})
RETURN mr.value as PERatio

# Response answer
The Price-to-Earnings (P/E) ratio of PTTEP on September 1, 2023, is 8.05.

# Expected answer
PTTEP's P/E ratio on September 1, 2023 is 8.05
```


[4] P. Kotiranta, M. Junkkari, and J. Nummenmaa, "Performance of graph and relational databases in complex queries," *Applied Sciences*, vol. 12, no. 13, 2022.

[5] J. Sequeda, D. Allemang, and B. Jacob, "A benchmark to understand the role of knowledge graphs on large language model's accuracy for question answering on enterprise sql databases," in *Proceedings of the 7th Joint Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, ser. GRADES-NDA '24. New York, NY, USA: Association for Computing Machinery, 2024.

[6] S. Zehra, S. F. M. Mohsin, S. Wasi, S. I. Jami, M. S. Siddiqui, and M. K.-U.-R. R. Syed, "Financial knowledge graph based financial report query system," *IEEE Access*, vol. 9, pp. 69 766–69 782, 2021.

[7] M. Besta, R. Gerstenberger, E. Peter, M. Fischer, M. Podstawski, C. Barthels, G. Alonso, and T. Hoefler, "Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries," *ACM Comput. Surv.*, vol. 56, no. 2, Sep. 2023.

[8] R. Angles, "A comparison of current graph database models," in *2012 IEEE 28th International Conference on Data Engineering Workshops*, 2012, pp. 171–177.

[9] N. Kertkeidkachorn, R. Nararatwong, Z. Xu, and R. Ichise, "FinKG: A core financial knowledge graph for financial analysis," in *2023 IEEE 17th International Conference on Semantic Computing (ICSC)*, 2023, pp. 90–93.

[10] P. Shi, R. Zhang, H. Bai, and J. Lin, "XRICL: Cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-SQL semantic parsing," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 5248–5259.

[11] Y. Ye, B. Hui, M. Yang, B. Li, F. Huang, and Y. Li, "Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 174–184.

[12] H. Li, J. Su, Y. Chen, Q. Li, and Z. Zhang, "Sheetcopilot: bringing software productivity to the next level through large language models," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2024.

[13] N. Kertkeidkachorn and R. Ichise, "An automatic knowledge graph creation framework from natural language text," *IEICE Transactions on Information and Systems*, vol. E101.D, no. 1, pp. 90–98, 2018.

[14] W. Wang, Y. Xu, C. Du, Y. Chen, Y. Wang, and H. Wen, "Data set and evaluation of automated construction of financial knowledge graph," *Data Intelligence*, vol. 3, no. 3, pp. 418–443, 09 2021.

[15] S. Elhammadi, L. V. Lakshmanan, R. Ng, M. Simpson, B. Huai, Z. Wang, and L. Wang, "A high precision pipeline for financial knowledge graph construction," in *Proceedings of the 28th international conference on computational linguistics*, 2020, pp. 967–977.

[16] A. Khan, "Knowledge graphs querying," *SIGMOD Rec.*, vol. 52, no. 2, pp. 18–29, Aug. 2023.

[17] R. J. Sholichah, M. Imrona, and A. Alamsyah, "Performance analysis of neo4j and mysql databases using public policies decision making data," in *2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. IEEE, 2020, pp. 152–157.

[18] X. Li, Z. Li, C. Shi, Y. Xu, Q. Du, M. Tan, J. Huang, and W. Lin, "AlphaFin: Benchmarking financial analysis with retrieval-augmented stock-chain framework," 2024. [Online]. Available: https://arxiv.org/abs/2403.12582

[19] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "BloombergGPT: A large language model for finance," 2023. [Online]. Available: https://arxiv.org/abs/2303.17564

[20] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[21] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: retrieval-augmented language model pre-training," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[22] B. Zhang, H. Yang, T. Zhou, M. Ali Babar, and X.-Y. Liu, "Enhancing financial sentiment analysis via retrieval augmented large language models," in *Proceedings of the Fourth ACM International Conference on AI in Finance*, ser. ICAIF '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 349–356.

[23] W. Yu, "Retrieval-augmented generation across heterogeneous knowledge," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, D. Ippolito, L. H. Li, M. L. Pacheco, D. Chen, and N. Xue, Eds. Hybrid: Seattle, Washington + Online: Association for Computational Linguistics, Jul. 2022, pp. 52–58.

[24] Z. Xu, M. J. Cruz, M. Guevara, T. Wang, M. Deshpande, X. Wang, and Z. Li, "Retrieval-augmented generation with knowledge graphs for customer service question answering," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 2905–2909.

[25] 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, no. 7, pp. 3580–3599, 2024.

[26] M. R. Parvez, W. Ahmad, S. Chakraborty, B. Ray, and K.-W. Chang, "Retrieval augmented code generation and summarization," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2719–2734.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:46 UTC from IEEE Xplore. Restrictions apply.