DS@GT at FinMMEval 2026 Task 2: FinNexus-Agentic Retrieval-Augmented Reranking Orchestration for Multilingual Financial Question Answering

FinMMEval at CLEF 2026

Dianze Liu[1], Sirui Li[2] and Shuyu Tian[1,*]

[1]Georgia Institute of Technology, Atlanta, GA, USA
[2]ByteDance Inc., San Jose, CA, USA

Abstract

Financial question answering and reasoning should not rely on a single information source. Financial knowledge is inherently heterogeneous, encompassing financial statements, market news, and multilingual discussions that offer diverse perspectives, interpretations, and insights on a given topic. Effectively aggregation of these sources and selectively identifying the most relevant information to answer questions remains a significant engineering challenge. In this work, we introduce an agentic framework that decomposes this complex task into manageable stages and addresses them incrementally. We propose a multi-stage pipeline that combines semantic retrieval with a specialized reranking module, alongside an LLM-driven planner capable of Python-based tool execution for accurate numerical analysis. The system is developed and evaluated on the first multilingual financial question answering dataset, PolyFiQA [1]. Experimental results demonstrate that our approach can generate high-quality, grounded responses to user queries.

Keywords

Question Answering, Reasoning, Agentic Workflow, Reranking, Multilingual Embedding

1. Introduction

Finance is inherently global. Investors, analysts, researchers, and regulators worldwide continuously consume and interpret information related to corporate performance and financial markets. Financial information derives from diverse sources, including quarterly financial statements, market reports, earnings calls, and multilingual news articles produced by journalists and institutions located in different countries. As the volume of financial information continues to grow rapidly, navigating this vast information landscape and identifying truly relevant insights has become increasingly challenging.

Human experts with strong financial domain knowledge are often capable of identifying key indicators and extracting meaningful signals from structured financial statements. However, processing unstructured information such as financial news requires reading through large volumes of articles to locate supporting evidence and contextual information. The challenge becomes even greater in multilingual settings, where relevant information may be distributed across different languages and reporting styles. Performing accurate reasoning across multilingual and multi-source financial information therefore demands not only domain expertise, but also the ability to synthesize and logically connect heterogeneous evidence [2].

To address these limitations, we propose an agentic financial reasoning workflow designed specifically for multilingual, multi-hop financial question answering. Recent research has shown that agentic systems can effectively decompose complex tasks into smaller subtasks, enabling more reliable reasoning, planning, and tool utilization. However, designing an effective agentic workflow for financial reasoning requires addressing several domain-specific challenges.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
*Corresponding author.
dliu450@gatech.edu (D. Liu); sirui.li@bytedance.com (S. Li); stian40@gatech.edu (S. Tian)
0009-0008-2651-3556 (D. Liu); 0000-0001-6444-651X (S. Tian)

Financial question answering frequently requires accurate computation of financial ratios, trend analysis, and manipulation of structured financial data. To address these requirements, we design a multi-stage agentic workflow that integrates multilingual semantic retrieval, reranking, and tool-augmented reasoning. Specifically, we employ multilingual vector embeddings [3] to align multilingual financial news into a unified retrieval space, enabling cross-language information access. Retrieved documents are further refined using a reranking module to improve relevance and reduce retrieval noise. We additionally equip the LLM agent with Python-based tool execution capabilities, enabling accurate numerical computation and structured financial analysis.

We developed and evaluated our system on PolyFiQA [1], the target benchmark for Task 2 of the FinMMEval Lab [4, 5]. As the first multilingual financial question answering benchmark. PolyFiQA contains both Easy and Expert subsets, where each query consists of a company-specific financial question, structured financial statements—including balance sheets, income statements, and cash flow statements—and multilingual news articles written in English, Chinese, Spanish, Japanese, and Greek. The benchmark evaluates the system's ability to generate grounded answers by synthesizing evidence across structured financial data and multilingual textual sources. Experimental results demonstrate that our agentic framework enables smaller and lower-cost LLMs to achieve competitive performance compared to significantly larger flagship models, while maintaining strong reasoning accuracy and grounded financial analysis capabilities.

2. Related Works

Recent advances in natural language processing and large language models (LLMs) have demonstrated promising capabilities in financial applications. Early domain-specific models such as FinBERT demonstrated the effectiveness of transformer-based architectures for financial sentiment analysis and classification tasks [6]. More advanced financial foundation models such as BloombergGPT further expanded these capabilities to include financial question answering, text generation, and domain-specific reasoning [2]. However, these systems remain limited when handling complex multilingual and multi-hop financial reasoning tasks. While models such as BloombergGPT can answer relatively straightforward financial questions, they are not specifically optimized for cross-document reasoning that requires synthesizing evidence across multiple languages and heterogeneous data sources. To evaluate these limitations, recent benchmarks such as MultiFinBen have introduced tasks requiring complex reasoning over multiple language inputs and multimodal documents [7]. Meanwhile, earlier models like FinBERT remain focused primarily on narrow downstream tasks.

General-purpose LLMs have recently demonstrated strong multilingual understanding and reasoning capabilities, including performance on multi-hop question answering tasks [8]. These advances suggest that modern LLMs possess the potential to address complex multilingual financial reasoning problems. Nevertheless, deploying large-scale proprietary LLMs in real-world financial applications introduces several practical challenges. First, computational and monetary costs remain significant. Financial analysis workflows are highly iterative, where users often ask numerous follow-up questions during investment research and corporate analysis. Continuously querying large flagship models can therefore become prohibitively expensive. Second, LLMs are known to generate hallucinated or ungrounded responses, especially when relying heavily on parametric knowledge rather than retrieved evidence [9]. In high-stakes financial settings, ensuring factual grounding and traceability is essential. Third, long-context reasoning remains challenging for many models, as important information may be lost or underutilized when processing large collections of financial reports and multilingual news articles simultaneously [10].

Conventional retrieval-augmented generation (RAG) [11] systems can retrieve relevant financial news articles based on semantic similarity. However, structured financial data such as balance sheets, income statements, and cash flow statements are not naturally suited for standard embedding-based retrieval methods [12]. Financial terminology often occupies closely related regions within embedding spaces, potentially reducing retrieval precision and introducing ambiguity. Furthermore, retrieved

evidence must remain highly relevant and accurate to support reliable financial reasoning.

3. System Architecture

The system designed for the task is named FinNexus. The core idea behind this system is to decompose the complex task of multilingual financial question answering into smaller, controllable, and interpretable subtasks. FinNexus leverages large language models together with the LangGraph orchestration framework [13] to coordinate intermediate reasoning steps and compose the final grounded response.

Before processing user queries, FinNexus performs an extensive preprocessing stage on multilingual financial news articles. News documents are first grouped by language and segmented into chunks of approximately 100 words or characters, depending on the language characteristics. To preserve contextual continuity and prevent important information from being lost due to sentence boundaries, adjacent chunks are generated with a 40% overlap. Both chunk size and overlap ratio are treated as tunable hyperparameters and can be adjusted depending on the downstream task and query complexity.

FinNexus constructs a unified semantic vector space using multilingual embedding models [14, 15]. One possible approach for multilingual retrieval is to translate all foreign-language news into English before generating embeddings. However, this introduces potential semantic drift during translation, particularly when reconstructing quotations or preserving subtle linguistic nuances from the original articles [16]. Such translation artifacts may lead to information loss or slight shifts in meaning. To mitigate this issue, FinNexus directly employs multilingual embeddings that align semantically similar content from different languages into a shared vector space [14, 17], allowing the system to preserve the original textual evidence while enabling cross-lingual retrieval.

All processed news chunks are stored in a vector database using Weaviate [18] due to its efficient retrieval performance, ease of integration, and native support for reranking pipelines. In addition to handling unstructured multilingual news, FinNexus must also accurately process structured financial statements. During preliminary experiments, we observed that LLMs—particularly lightweight models—often struggled to interpret financial tables containing extensive formatting elements such as lines, separators, and aligned columns . Although these structures improve readability for humans, they introduce unnecessary tokens and potential ambiguity for language models.

To address this limitation, FinNexus removes non-essential table formatting and converts financial statements into simplified textual representations containing only meaningful labels and numerical values. Column headers naturally preserve structural relationships without requiring additional formatting symbols. This preprocessing significantly improves model comprehension while also reducing token consumption. On average, the system reduces approximately 3,000 tokens per query, as raw numerical and textual financial information accounts for only around 15% of the original formatted document. The processed financial statements are subsequently transformed into structured formats such as dictionaries or JSON objects, enabling easier programmatic access and reasoning.

FinNexus categorizes financial questions into two primary reasoning types. The first category focuses on extracting insights primarily from financial statements and subsequently using news articles as supporting evidence. The second category focuses on identifying relevant information from news articles first, followed by financial statement analysis to support or validate the conclusion. This design closely aligns with the structure of the PolyFiQA benchmark, where the Easy subset emphasizes financial-statement-centered reasoning, while the Expert subset requires more complex news-driven multi-hop reasoning [19].

The routing process is handled dynamically by the LLM agent without additional fine-tuning. Through carefully engineered prompting strategies, even lightweight models are capable of reliably identifying the appropriate reasoning workflow based on the user query. This routing mechanism enables FinNexus to generalize across different financial reasoning patterns while maintaining relatively low computational cost.

To support multilingual news reasoning, we develop a dedicated retrieval subsystem incorporating several retrieval-enhancement techniques. First, FinNexus selectively retrieves news collections as-

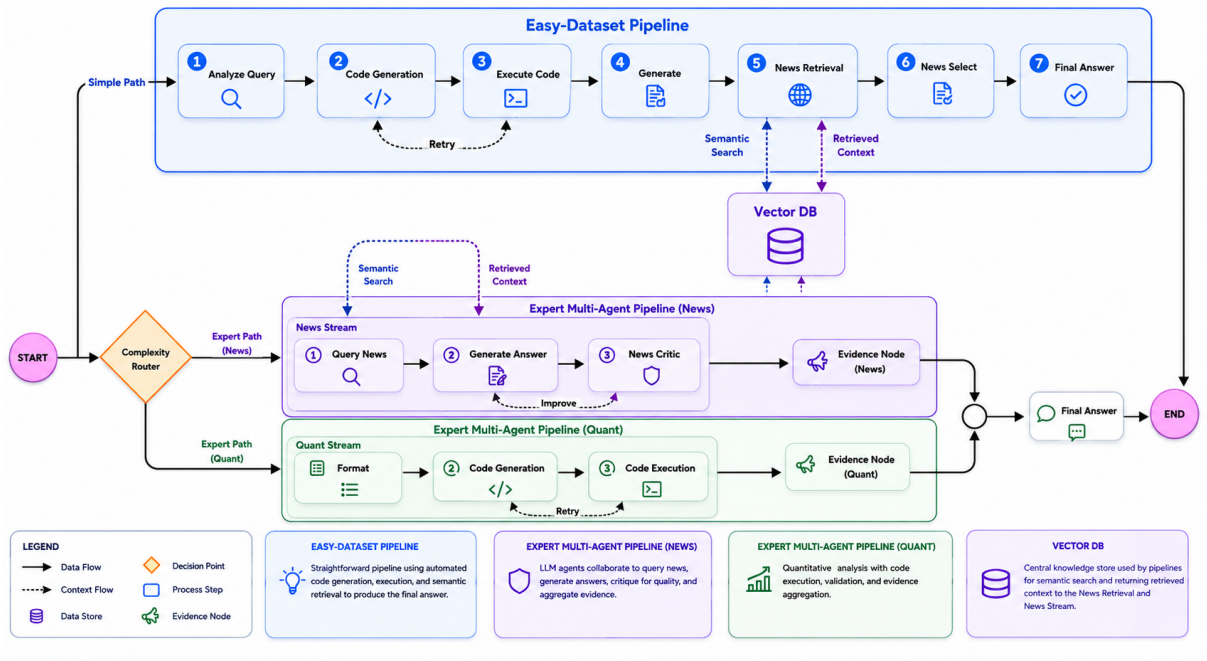

Figure 1: FinNexus Agent Orchestration Pipeline

sociated with the queried company, ensuring that retrieval remains focused on relevant information rather than being overwhelmed by unrelated articles. Second, the system enriches retrieval queries by incorporating intermediate findings extracted from financial statements, enabling the retrieval process to become progressively more context-aware. Experimental observations show that this iterative query enrichment significantly improves retrieval precision and downstream answer quality.

FinNexus further employs hybrid retrieval [20, 21], combining semantic vector search with traditional lexical retrieval methods such as BM25 [22]. In practice, we assign a relatively high weight to semantic similarity (approximately 0.7–0.9) while maintaining a smaller BM25 contribution. This configuration provides strong semantic matching while preserving important keyword-level precision.

To further improve retrieval quality, FinNexus integrates reranking models based on transformer architectures. Pure semantic retrieval may return contextually related passages that are not directly useful for answering the question. To address this issue, we employ the Voyage reranker [23] integrated within Weaviate, enabling efficient relevance scoring without requiring additional infrastructure for GPU-intensive reranking computations. Empirical experiments indicate that retrieving a relatively larger candidate pool (e.g., top 30 documents) before reranking substantially improves retrieval quality. After reranking, only the top three most relevant passages are forwarded to downstream reasoning components.

To improve robustness and reduce noisy evidence propagation, FinNexus additionally incorporates a judging agent responsible for critically evaluating retrieved evidence before inclusion in the final reasoning stage. Prompt engineering plays a crucial role in balancing the behavior of this judging component. Without carefully designed prompts, the judging agent may become overly conservative by rejecting nearly all retrieved evidence, or overly permissive by accepting irrelevant documents indiscriminately. Through iterative prompt refinement, we achieve a balanced filtering strategy that improves overall answer grounding.

For financial statement reasoning, FinNexus accesses the preprocessed structured financial data stored as JSON objects. Since financial reports contain multiple statement types—including balance sheets, income statements, and cash flow statements—the system first determines which statement is relevant to the user query before accessing the data. Restricting access to only the necessary financial statement reduces confusion caused by semantically similar terminology across different reports.

During experimentation, we observed that lightweight LLMs occasionally produced incorrect cal-

culations or referenced incorrect numerical values when performing direct arithmetic reasoning. To mitigate this issue, FinNexus delegates numerical computation to executable Python code rather than relying solely on natural language reasoning. Specifically, the LLM generates Python programs that access the structured financial data and compute the required financial metrics according to the user query. The generated code is then executed within a Python runtime environment.

FinNexus utilizes LangGraph to maintain execution states, intermediate outputs, and error traces throughout the reasoning workflow. If execution errors occur, the generated code and corresponding error messages are returned to the code-generation component for iterative correction and regeneration. This retry mechanism substantially improves computational reliability while reducing numerical reasoning errors commonly observed in direct LLM outputs.

After both retrieval and numerical analysis stages are completed, FinNexus proceeds to the final answer synthesis stage. At this point, the system has accumulated structured evidence from financial statement analysis together with supporting multilingual news evidence. Before generating the final response, a verification agent reviews all retrieved and computed information to retain only evidence directly relevant to the user query. This stage adopts an "LLM-as-a-Judge" paradigm [24] to evaluate the consistency, relevance, and grounding quality of intermediate findings.

Finally, the answer generation module synthesizes the verified evidence into a coherent grounded response tailored to the user query. By decomposing the overall reasoning process into modular retrieval, computation, verification, and synthesis stages, FinNexus achieves both flexibility and robustness. Experimental results demonstrate that the system can effectively answer multilingual financial questions while providing grounded explanations supported by both financial statements and multilingual news evidence, extending beyond the constraints of benchmark datasets alone.

4. Experiment and Setup

To comprehensively evaluate the performance of the Finnexus system, we designed an experimental framework across multiple dimensions using two distinct datasets. The first dataset consists of the PolyFiQA [1] training data. The second is a dedicated testing dataset containing instances unseen by the system during development, with the ground-truth answers held out for evaluation purposes.

Our evaluation rigorously assesses three primary aspects of the Finnexus system:

1. **Textual Quality and Quotation Accuracy:** We utilize the ROUGE-1 score [25] to evaluate general answer quality. Additionally, we measure the system's quotation ability by verifying whether it can accurately cite the specific financial news provided in the ground-truth data.
2. **Quantitative and Calculation Ability:** To isolate and measure the system's ability to accurately deliver financial computations, we extract generated metrics such as cash flows, financial ratios, and revenues. We then verify if these values match the ground-truth answers. Crucially, we exclude numbers directly quoted from the source text, as reproducing these relies on copy-and-paste extraction rather than true computational ability.
3. **Token Consumption and Financial Cost:** Given that operational cost is a critical constraint for deploying agentic systems in production—where users heavily favor cost-efficient methods that maintain high performance—we track total token consumption and actual monetary costs.

For benchmarking, we compare a flagship large language model (LLM) against the Finnexus system, which utilizes a lighter, faster-responding variant of the model. Finally, data points affected by API errors were excluded from the final analysis. These errors occurred due to rate limits imposed by intensive API usage during rapid, short-term data generation. Excluding these instances ensures that transient API failures do not artificially skew the evaluation of the system's actual performance.

For the official PolyFiQA Task 2 evaluation, we submitted a single run to the evaluation server. The submitted run corresponds to the FinNexus agentic system described in this paper, powered by Gemini 3.1 Flash-Lite with the complete retrieval, reasoning, and financial calculation pipeline presented in Section 3. All official leaderboard results reported in this paper are based on this single submitted run.

Table 1
ROUGE-1 Score Results on the Easy and Expert Splits for Gemini 3.5 and the FinNexus Agentic System

Model	Split	Count	Precision	Recall	F1
Gemini 3.5 Flash-Lite Baseline	Easy	76	**0.5042**	**0.5035**	**0.4671**
Gemini 3.5 Flash-Lite Baseline	Expert	76	0.2775	0.4474	0.3246
Gemini 3.5 Flash-Lite Baseline	Overall	152	**0.3908**	**0.4754**	**0.3959**
FinNexus	Easy	71	0.3748	0.4270	0.3694
FinNexus	Expert	69	**0.3114**	**0.4821**	**0.3666**
FinNexus	Overall	140	0.3436	0.4542	0.3680

5. Results and Discussion

This section presents the evaluation results across multiple metrics to compare the performance of the FinNexus agentic system (powered by Gemini 3.5 Flash-Lite) against direct API calls to the baseline flagship language model. The evaluation benchmarks the models across text quality, numerical calculation accuracy, token consumption, and generalization capabilities on the unseen dataset.

5.1. ROUGE-1 Score Evaluation

Table 1 details the ROUGE-1 performance metrics for the direct Gemini 3.5 Flash-Lite baseline API calls versus the FinNexus agentic system evaluated on the PolyFiQA dataset splits.

Due to transient API rate-limit failures during inference, the FinNexus system successfully generated valid outputs for 140 of the 152 evaluation instances, whereas the Gemini 3.5 baseline completed all 152 instances. The API failures were infrastructure-related rather than model-related, and we have no evidence that the excluded samples were systematically associated with either the Easy or Expert question types. Nevertheless, because the missing instances were not explicitly controlled to preserve the Easy/Expert distribution, a small evaluation bias cannot be completely ruled out. Therefore, the head-to-head comparison reported in Table 1 should be interpreted with this limitation in mind.

The baseline model demonstrates superior performance on the Easy split. However, this discrepancy is primarily driven by answer length and quotation behavior rather than structural generation failures. Direct Gemini 3.5 baseline calls tend to naively copy and paste large segments of financial news as references, yielding an average answer length of 74.84 words. Conversely, the FinNexus system generates more concise answers, averaging 54.96 words. This conservative behavior is driven by the system's "LLM-as-a-Judge" node, which rigorously filters out source news that is not highly relevant to the user query. While this strict filtering reduces the word count—and subsequently penalizes token-matching metrics such as ROUGE-1 on simple tasks—it prevents the inclusion of extraneous information.

On the Expert split, where deep reasoning is required, the FinNexus system significantly outperforms the baseline flagship model despite utilizing a lighter core engine. The agentic workflow forces the system to systematically parse related news and complex financial statements. Crucially, FinNexus achieves a noticeably higher Recall score (0.4821) on the Expert split, demonstrating that the agentic workflow successfully captures key domain-specific information that a standard zero-shot prompt misses. These results validate that a well-designed agentic architecture can overcome the limitations of smaller foundation models, delivering superior performance on complex financial reasoning tasks while minimizing operational costs.

5.2. Numerical Accuracy Evaluation

In this section, we evaluate the system's quantitative performance by analyzing the accuracy of generated numerical values involving calculations, such as financial ratios, incomes, revenues, and spending metrics. As noted previously, numbers directly quoted from the source text were excluded from this

Table 2
Tier Accuracy Results at 1% Tolerance for Gemini 3.5 Flash-Lite and the FinNexus Agentic System

Model	Split	Accuracy
Gemini 3.5 Baseline	Easy	31.87%
Gemini 3.5 Baseline	Expert	27.02%
FinNexus	Easy	**36.10%**
FinNexus	Expert	**31.64%**

analysis to isolate the systems' actual computational capabilities rather than their retrieval and extraction accuracy.

Table 2 presents the numerical accuracy results under a strict 1% error tolerance threshold. This margin was explicitly introduced to account for minor rounding discrepancies and variation in numerical presentation formats (e.g., decimal places or percentage signs) that would otherwise cause a functionally correct generated answer to mismatch the ground-truth target. Even under this robust evaluation metric, the results demonstrate that the FinNexus system consistently outperforms the baseline flagship Gemini 3.5 model across both the Easy and Expert splits.

This performance gain is primarily attributed to the architectural design of FinNexus, which incorporates a sandboxed Python code execution environment. Instead of relying on the underlying autoregressive next-token probability distribution to approximate mathematical results—a mechanism prone to calculation errors and hallucinations—FinNexus generates executable Python functions tailored to the user's specific request. By shifting the computational burden from linguistic prediction to an exact execution environment, the system effectively eliminates basic math errors. Furthermore, this programmatic approach forces a more structured parsing of the underlying financial data, enabling the system to interpret complex financial queries with higher fidelity and produce highly accurate quantitative answers.

5.3. Token Consumption and Financial Cost

In this section, we evaluate token consumption and estimate the operational costs of both methods to assess the financial viability of the FinNexus system on PolyFiQA tasks. Given that FinNexus achieves comparable accuracy while providing robust answer grounding through news articles and financial statements, quantifying its economic efficiency is essential for determining its practicality in production environments.

Table 3 presents a detailed breakdown of the average token usage and monetary cost per query. Although FinNexus orchestrates multiple internal LLM calls—including query understanding, financial news chunk filtering, and Python code generation—the overall input and output token consumption remains remarkably low. This efficiency is primarily enabled by three key architectural design choices.

First, FinNexus employs dense vector embeddings to index and store financial text chunks. While overlapping text fragments naturally arise during document chunking, the relatively low cost of embedding models substantially reduces both initialization and retrieval overhead. Second, instead of passing the entire news corpus or all available fragments—which average approximately 100 chunks per query—FinNexus selectively forwards only the top 30 highly relevant chunks for reranking. Consequently, only 30% of the candidate data undergoes computationally expensive reranking, preserving a broad retrieval space while significantly reducing token overhead.

Moreover, reranking itself remains relatively inexpensive compared to premium flagship models. The lightweight reranker costs approximately $0.025 per million tokens, while the premium reranker costs roughly $0.05 per million tokens. As a result, selecting the highest-performing reranker introduces only a negligible increase in operational expense.

Finally, FinNexus leverages a significantly cheaper and faster core model variant for its internal agentic loops, enabling scalable multi-step reasoning without incurring the prohibitive costs typically

Table 3
Average Token Usage and Estimated Cost per Query

System Component	Token Usage	Pricing / Million Tokens	Estimated Cost
Baseline Flagship (LLM Input)	11,939	$1.50	$0.01791
Baseline Flagship (LLM Output)	1,300 (Approx.)	$9.00	$0.01170
Baseline Flagship (Total)	–	–	**$0.02961**
FinNexus (LLM Input)	3,634	$0.25	$0.00091
FinNexus (LLM Output)	1,211	$1.50	$0.00182
FinNexus (Embedding)	15,521	$0.15	$0.00233
FinNexus (Reranker)	3,600 (Approx.)	$0.05	$0.00018
FinNexus (Total)	–	–	**$0.00524**

associated with repeated flagship model interactions.

Overall, the FinNexus system operates at less than 19% of the total cost incurred by direct flagship model calls ($0.0052 versus $0.0296 per query). This substantial reduction demonstrates that an agentic framework can generate high-quality, grounded financial answers while operating under strict budget constraints, making the system highly suitable for real-world enterprise deployment.

5.4. Evaluation on Unseen Holdout Dataset

To evaluate the generalization capabilities of FinNexus, the system was benchmarked on a distinct holdout dataset containing out-of-distribution queries that were completely unseen during the system's development. This testing isolates how effectively the multi-agent architecture generalizes to varied, open-domain financial questions without tailored prompt engineering. On this holdout data, FinNexus achieved a ROUGE-1 Precision of 0.2572, a Recall of 0.3888, and a baseline ROUGE-1 F1 score of 28.53%.

While these metrics demonstrate robust algorithmic consistency, a marginal performance drop was observed compared to the primary evaluation splits. This variance is attributable to two primary infrastructure and evaluation constraints. First, due to rate limit overloads stemming from rapid concurrent requests during this specific testing phase, the evaluation pipeline could not cleanly isolate and exclude transient API connection errors, meaning failed API responses remained in the final calculation. Second, the holdout evaluation protocol enforces a strict restriction limiting all ground-truth target answers to a maximum of 100 words. FinNexus, by design, prioritizes the architectural integrity of its textual citations by extracting complete, unedited source sentences rather than lossy summarization or premature sentence chunking. This design choice optimizes real-world readability and financial auditability; however, within a strict length-penalized evaluation framework, it caused long, accurate evidence strings to be truncated midway through processing. This artificially lowered the text-matching scores despite the underlying factual correctness of the generated content.

6. Conclusion

In this work, we presented FinNexus, an agentic orchestration solution designed to answer complex financial questions across multi-source financial reports and news, as framed by the PolyFiQA dataset. Experimental results demonstrate that FinNexus achieves significant cost advantages compared to relying strictly on premium large language models. Furthermore, the system exhibits a robust capability for accurate, grounded numerical calculations. While the raw accuracy metrics leave room for improvement, qualitative analysis indicates that the system often generates relevant auxiliary calculations to support its reasoning, leading to an overall improvement in synthesizing ratios, revenues, and numerical answers.

Our approach validates that embedding human domain knowledge—by decomposing complex financial queries into modular sub-tasks and assembling them via LangGraph alongside lightweight

language models—yields superior performance, particularly on specialized expert datasets. Additionally, FinNexus offers exceptional flexibility; its architecture allows for seamless customization and tuning through diverse prompting methodologies or by dynamically adding and removing sub-agents to align outputs with specific downstream goals.

Despite these advantages, the system exhibits vulnerability to unreliable API calls under high concurrent request volumes, which occasionally led to downstream errors during evaluation. In production settings, this limitation can be mitigated by surfacing explicit error messages to the user and triggering automated retries to ensure accurately grounded answers. Overall, FinNexus offers a promising, cost-effective, and highly flexible framework for automated financial analysis and multi-agent reasoning.

7. Future Work

During the tuning and optimization of the FinNexus system, we observed that text retrieval from the vector database is frequently executed without direct, independent evaluation. This occurs primarily because downstream training sets or sample instances often lack explicit ground-truth source document mappings. However, since the retrieval stage dictates the boundary of available context and heavily influences the quality of the final generated answer, establishing a systematic evaluation framework for retrieval remains a critical next step. Implementing granular retrieval metrics is essential to guide architectural scaling, providing a data-driven justification when introducing new pipeline tools such as neural rerankers or LLM-as-a-Judge filtering nodes.

To address this challenge, our future work will focus on developing a systematic methodology for constructing synthetic ground-truth retrieval sets. This pipeline will operate inversely from the ground-truth answers: using a seed answer, the system will reverse-engineer search queries to isolate the underlying source segments within the database. A highly confident, flagship LLM will act as a primary filter to automatically preserve definitive context matches. To maintain academic rigor and handle ambiguous or edge-case context segments, an ensemble approach leveraging advanced frontier models or human-in-the-loop annotations will be integrated to capture subtle but highly relevant background data.

With a structured golden retrieval dataset established, alternative vector retrieval, chunking, and embedding strategies can be scientifically audited. We plan to measure retrieval performance using standard rank-aware Information Retrieval metrics, including Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG). Utilizing these benchmarks will allow us to mathematically optimize our pipeline's top-K context windows, balancing high downstream factual recall against total operational token costs.

8. Acknowledgments

We thank the Data Science at Georgia Tech (DS@GT) CLEF competition group for providing organizational structure and research guidance.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4.1 and Gemini 3.5 Flash in order to: Grammar and spelling check. Further, the author(s) used GPT Image 2 for figures 1 in order to: Generate system flow images. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] Zhuohan Xie et al. *The CLEF-2026 FinMMEval Lab: Multilingual and Multimodal Evaluation of Financial AI Systems*. arXiv preprint arXiv:2602.10886. cs.CL. Feb. 2026. DOI: 10.48550/arXiv.2602.10886.

[2] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. "FinGPT: Open-Source Financial Large Language Models". In: *Scientific Data* (2023). DOI: 10.2139/ssrn.4489826. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4489826.

[3] Jinhyuk Lee et al. "Gemini Embedding: Generalizable Embeddings from Gemini". In: *arXiv* (2025). DOI: 10.48550/arxiv.2503.07891. URL: https://arxiv.org/abs/2503.07891.

[4] Zhuohan Xie et al. "Overview of FinMMEval 2026: Multilingual and Multimodal Financial Evaluation". In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026). Jena, Germany: Springer Lecture Notes in Computer Science LNCS, Sept. 2026.

[5] Zhuohan Xie et al. "Overview of the FinMMEval 2026 Task 2: Financial Question Answering and Summarization". In: *CLEF 2026 Working Notes*. CEUR Workshop Proceedings. Jena, Germany: CEUR-WS.org, Sept. 2026.

[6] Tingsong Jiang and Qingyun Zeng. "Financial sentiment analysis using FinBERT with application in predicting stock movement". In: *arXiv* (2023). DOI: 10.48550/arxiv.2306.02136. URL: https://arxiv.org/abs/2306.02136.

[7] Xueqing Peng et al. *MultiFinBen: Benchmarking Large Language Models for Multilingual and Multimodal Financial Application*. 2025. DOI: 10.48550/arXiv.2506.14028. arXiv: 2506.14028 [cs.CL]. URL: https://arxiv.org/abs/2506.14028.

[8] Simon Ott et al. "ThoughtSource: A central hub for large language model reasoning data". In: *Scientific Data* 10 (2023). DOI: 10.1038/s41597-023-02433-3. URL: https://www.nature.com/articles/s41597-023-02433-3.

[9] Michail Dadopoulos et al. "Metadata-Driven Retrieval-Augmented Generation for Financial Question Answering". In: *arXiv* (2025). DOI: 10.48550/arxiv.2510.24402. URL: https://arxiv.org/abs/2510.24402.

[10] Nelson F. Liu et al. "Lost in the Middle: How Language Models Use Long Contexts". In: *Transactions of the Association for Computational Linguistics* 12 (2024), pp. 157–173. DOI: 10.1162/tacl_a_00638. URL: https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00638/119630/Lost-in-the-Middle-How-Language-Models-Use-Long.

[11] Patrick Lewis et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020, pp. 9459–9474.

[12] Antonio Jimeno Yepes et al. "Financial Report Chunking for Effective Retrieval Augmented Generation". In: *arXiv* (2024). DOI: 10.48550/arxiv.2402.05131. URL: https://arxiv.org/abs/2402.05131.

[13] Harrison Chase and LangChain Inc. *LangGraph: Orchestrating Stateful, Multi-Agent Applications with LLMs*. 2024. URL: https://github.com/langchain-ai/langgraph.

[14] Nanyun Lee et al. "CORAL: Adaptive Retrieval Loop for Culturally-Aligned Multilingual RAG". In: *arXiv* (2026). URL: https://arxiv.org/abs/2604.25676.

[15] Wuttikorn Ponwitayarat et al. "SEA-BED: Southeast Asia Embedding Benchmark". In: *arXiv* (2025). DOI: 10.48550/arxiv.2508.12243. URL: https://arxiv.org/abs/2508.12243.

[16] Makiko Kimura-Ida. "Cross-lingual Comparison of Research Funding Projects with Multilingual Sentence-BERT: Evidence from KAKENHI, NIH, NSF, and UKRI". In: *arXiv* (2026). URL: https://arxiv.org/abs/2604.27315.

[17] Zihan Wang, Steven Feng, and John Lee. “DLIR: Spherical Adaptation for Cross-Lingual Knowledge Transfer of Sociological Concepts Alignment”. In: *arXiv* (2024). URL: https://openresearch.surrey.ac.uk/main/artifacts/digital_artifact/7e5fca9b-df86-4f96-be29-d5c4883ef593.

[18] Bob Ballsteijn and Weaviate BV. *Weaviate: An Open-Source, Cloud-Native Vector Search Engine for Stateful Applications*. 2024. URL: https://github.com/weaviate/weaviate.

[19] Yixuan Tang and Qiang Yang. “MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries”. In: *arXiv* (2024). DOI: 10.48550/arxiv.2401.15391. URL: https://arxiv.org/abs/2401.15391.

[20] Vladimir Karpukhin et al. “Dense Passage Retrieval for Open-Domain Question Answering”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://aclanthology.org/2020.emnlp-main.550/.

[21] Yi Luan et al. “Sparse, Dense, and Attentional Representations for Text Retrieval”. In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 329–345. DOI: 10.1162/tacl_a_00369. URL: https://aclanthology.org/2021.tacl-1.20/.

[22] Stephen Robertson and Hugo Zaragoza. “The Probabilistic Relevance Framework: BM25 and Beyond”. In: *Foundations and Trends® in Information Retrieval* 3.4 (2009), pp. 333–389. DOI: 10.1561/1500000019. URL: https://nowpublishers.com/article/Details/INR-019.

[23] Yiming Cui and Voyage AI Team. *Voyage Rerankers: High-Context Cross-Encoders for Retrieval-Augmented Generation*. 2024. URL: https://docs.voyageai.com/docs/reranker.

[24] Lianmin Zheng et al. “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena”. In: *arXiv* (2023). DOI: 10.48550/arxiv.2306.05685. URL: https://arxiv.org/abs/2306.05685.

[25] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. 2004, pp. 74–81.