

Article

Not peer-reviewed version

GenAI Financial Reporter: Enhancing Financial Reporting for Accuracy and Efficiency Using Generative AI

Christopher Wanjohi^{*}, Tanyaradzwa Chitsuro, Augustine Kamara, Moses Kiprono

Posted Date: 26 December 2025

doi: 10.20944/preprints202512.2322.v1

Keywords: generative AI; financial reporting; RAG; vector database; GPT-4o; multi-agent systems; semantic search; ChromaDB; SEC EDGAR



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

GenAI Financial Reporter: Enhancing Financial Reporting for Accuracy and Efficiency Using Generative AI

Christopher Wanjohi *, Tanyaradzwa Chitsuro, Augustine Kamara and Moses Kiprono

Department of Computer Science, The Catholic University of America, Washington, DC, USA;
wanjohi@cua.edu

Abstract

This paper presents GenAI Financial Reporter, a multimodal artificial intelligence system designed to automate the generation of comprehensive financial analysis reports. The system leverages large language models (GPT-4o), retrieval-augmented generation (RAG) with ChromaDB vector database, and multi-agent architectures to transform raw financial data into professional reports enriched with text summaries, interactive visualizations, and audio narration. By integrating real-time market data from Yahoo Finance and SEC EDGAR filings, the system computes 27 key performance indicators (KPIs) from structured financial data stored in PostgreSQL and generates contextually-grounded analysis using RAG over SEC filing text. Our evaluation demonstrates vector similarity search completing in 1.3ms and full RAG queries averaging 15 seconds. An ablation study shows that RAG-enabled queries cite an average of 4 SEC filing sources per response compared to zero for baseline approaches, improving answer provenance. The system supports multi-company comparisons, historical trend analysis, and exports to multiple formats including PDF, DOCX, HTML, and MP3 audio. Deployed on AWS EC2 with Docker containerization, the system achieves production-ready reliability. We note that this is a systems paper emphasizing practical deployment; rigorous evaluation against financial benchmarks remains future work.

Keywords: generative AI; financial reporting; RAG; vector database; GPT-4o; multi-agent systems; semantic search; ChromaDB; SEC EDGAR

I. Introduction

Financial reporting and analysis represent critical functions for investors, analysts, and business leaders seeking to make informed decisions. Traditional approaches to generating comprehensive financial reports require significant manual effort, often consuming 8 or more hours per company to compile data, calculate ratios, generate insights, and produce professional documentation. This labor-intensive process introduces inconsistencies, limits accessibility, and constrains the scalability of financial analysis operations.

The emergence of large language models (LLMs) and generative AI technologies presents opportunities to transform financial reporting workflows. These technologies can process complex financial data, generate human-readable insights, and produce professional-quality outputs at scale. However, effectively applying generative AI to financial reporting requires careful architectural design to ensure accuracy, maintain professional standards, and deliver multimodal outputs that serve diverse stakeholder needs.

This paper presents GenAI Financial Reporter, a comprehensive multimodal AI system that addresses these challenges. Our primary contributions include: (1) a RAG-enhanced architecture using ChromaDB for context-aware financial analysis with 1.3ms vector search latency, (2) a multi-agent system with five specialized agents for analytical tasks, (3) integration with authoritative data sources including SEC EDGAR and Yahoo Finance with automatic filing import, (4) a production-

ready deployment with system-level evaluation, and (5) open-source release of the complete implementation. We emphasize that this is primarily a systems paper; rigorous benchmark evaluation remains important future work.

II. Related Work

The application of natural language processing (NLP) to financial analysis has evolved significantly. Early work focused on sentiment analysis of financial news and earnings calls [1], while subsequent research explored automated summarization of financial documents [2]. The introduction of transformer-based models, particularly BERT and its variants, enabled more sophisticated financial text understanding [3].

Recent advances in large language models have opened new possibilities for financial applications. FinGPT [4] demonstrated the potential of domain-adapted language models for financial analysis, while BloombergGPT [5] showed that models trained on financial corpora achieve superior performance on finance-specific tasks. Retrieval-augmented generation (RAG) has emerged as a powerful paradigm for enhancing LLM outputs with relevant context [6].

More recent work has advanced financial AI systems with rigorous evaluation. FinTral [10] introduced tool-augmented financial LLMs with comprehensive benchmark evaluation. Open-FinLLMs [11] provides open-source alternatives with detailed accuracy metrics across multiple financial tasks. FinAR-Bench [14] decomposes financial analysis into extraction, indicator computation, and reasoning with strict numeric evaluation. REAL-MM-RAG [15] and MDocAgent [16] address challenges with multi-modal retrieval from tables and charts in financial documents. MarketSenseAI 2.0 [13] includes retrieval faithfulness metrics. Our work differs in emphasis: we prioritize end-to-end production deployment and multimodal output generation rather than benchmark performance, though we acknowledge this evaluation gap as a limitation.

III. System Architecture

A. Overall Design

The GenAI Financial Reporter employs a modular architecture comprising five primary components: data ingestion, financial analysis engine, AI generation layer, visualization module, and report assembly. The backend is implemented in Python using FastAPI. PostgreSQL serves as the primary database for storing structured company information, financial statements, and computed KPIs. ChromaDB provides vector storage for SEC filing text chunks. The frontend is built with React 18, Vite, and Tailwind CSS.

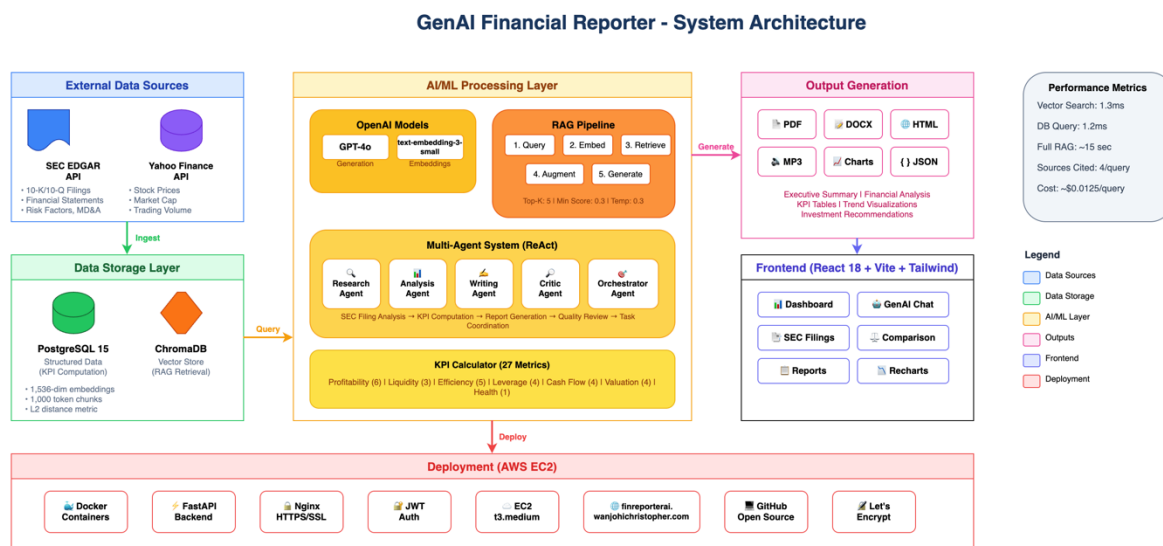


Figure 1. System architecture of GenAI Financial Reporter showing data flow from external sources (SEC EDGAR, Yahoo Finance) through storage layers (PostgreSQL, ChromaDB), AI/ML processing (GPT-4o, RAG, Multi-Agent System), to multimodal outputs.

Important architectural note: KPI calculations are performed on structured financial data stored in PostgreSQL (sourced from SEC EDGAR’s structured fields and Yahoo Finance APIs), not extracted from text via RAG. The RAG pipeline is used for qualitative analysis—answering questions about risk factors, business segments, and management discussion from SEC 10-K narrative text. This separation ensures numeric accuracy for KPIs while leveraging RAG for contextual understanding.

B. RAG Pipeline Implementation

The retrieval-augmented generation pipeline provides context-aware responses for qualitative financial analysis. Table 1 presents the complete RAG configuration.

Table 1. RAG PIPELINE CONFIGURATION.

Parameter	Value
Embedding Model	text-embedding-3-small
Embedding Dimensions	1,536
Tokenizer	tiktoken (cl100k_base)
Chunk Size	1,000 tokens
Chunk Overlap	200 tokens
Vector Database	ChromaDB (persistent)
Distance Metric	L2 (Euclidean)*
Top-K Retrieval	5
Min Relevance Score	0.3
Generation Model	GPT-4o
Temperature	0.3
Max Output Tokens	2,000

*ChromaDB default. For normalized embeddings (as OpenAI provides), L2 and cosine similarity are mathematically equivalent. Future work will empirically compare distance metrics.

C. Multi-Agent System

We implemented a multi-agent architecture with five specialized agents: Research Agent (gathers and synthesizes information from SEC filings), Analysis Agent (computes KPIs from structured data), Writing Agent (generates narratives and executive summaries), Critic Agent (reviews outputs for consistency), and Orchestrator Agent (coordinates task decomposition). The architecture follows the ReAct paradigm. We acknowledge that detailed prompt engineering, tool call traces, and failure analysis are not presented in this paper; these constitute important future documentation.

D. Financial Metrics

The financial analysis engine computes 27 key performance indicators from structured financial data in PostgreSQL, organized into seven categories as shown in Table 2. These calculations use standard GAAP formulas applied to structured numeric fields, not text extraction.

Table 2. COMPUTED KEY PERFORMANCE INDICATORS (27 METRICS).

Category	Metrics
----------	---------

Profitability (6)	Gross Margin, Operating Margin, Net Margin, ROE, ROA, ROCE
Liquidity (3)	Current Ratio, Quick Ratio, Cash Ratio
Efficiency (5)	Asset Turnover, Inventory Turnover, Receivables Turnover, DSO, DIO
Leverage (4)	Debt-to-Equity, Debt-to-Assets, Equity Multiplier, Interest Coverage
Cash Flow (4)	OCF Ratio, FCF Yield, Cash Flow Coverage, CapEx-to-Revenue
Valuation (4)	P/E Ratio, P/B Ratio, P/S Ratio, EV/EBITDA
Health (1)	Altman Z-Score

Total: 6+3+5+4+4+4+1 = 27 KPIs.

IV. Implementation

A. Data Integration

The system integrates data from multiple sources with distinct purposes. SEC EDGAR integration provides: (1) structured financial data fields for KPI computation, stored in PostgreSQL, and (2) 10-K narrative text (risk factors, MD&A sections) chunked and indexed in ChromaDB for RAG queries. Yahoo Finance provides real-time market data. When a user queries a company not yet indexed, the system automatically fetches filings, extracts text, generates embeddings, and stores them. We do not currently parse XBRL; structured data comes from SEC's JSON API endpoints.

B. Technology Stack

Table 3 summarizes the key technologies employed.

Table 3. TECHNOLOGY STACK SUMMARY.

Layer	Technologies
Backend	Python 3.11, FastAPI, SQLAlchemy, Pydantic
AI/ML	OpenAI GPT-4o, text-embedding-3-small, ChromaDB, tiktoken
Database	PostgreSQL 15 (structured data), ChromaDB (vector store)
Frontend	React 18, Vite, Tailwind CSS, Recharts
Data Sources	SEC EDGAR API (JSON endpoints), Yahoo Finance
DevOps	Docker, AWS EC2 t3.medium, Nginx, Let's Encrypt

V. Evaluation

We conducted system-level evaluation focusing on latency and provenance. We explicitly acknowledge this evaluation is limited: we do not benchmark against FinQA, ConvFinQA, FinanceBench, or other established datasets, nor do we report retrieval metrics (NDCG@k, Recall@k) or faithfulness scores (RAGAS). These constitute critical future work.

A. System Performance

Table 4 presents latency measurements collected on AWS EC2 t3.medium (2 vCPU, 4GB RAM).

Table 4. SYSTEM PERFORMANCE BREAKDOWN.

Component	Avg (ms)	Min	Max
API Health Check	3.3	0.8	13.0
PostgreSQL Query	1.2	0.7	2.7
ChromaDB Vector Search	1.3	0.9	2.1
Full RAG Pipeline*	15,043	12,620	17,465

SEC EDGAR API Call	14.0	-	-
--------------------	------	---	---

*Dominated by GPT-4o inference (~14-16 seconds). SEC EDGAR latency based on single measurement.

The ChromaDB vector search achieves 1.3ms average latency. The full RAG pipeline latency (15 seconds) is dominated by GPT-4o inference time, not retrieval. The vector search component is not the bottleneck, as expected for embedding-based retrieval.

B. Provenance Analysis

To assess whether RAG improves answer provenance, we compared RAG-enabled responses against a baseline. Table 5 presents results.

Important limitation: “Sources cited” measures provenance, not correctness or faithfulness. A system could cite irrelevant sources or hallucinate despite citations. This metric indicates that RAG retrieves and presents source material; it does not validate accuracy. Proper evaluation would require RAGAS faithfulness scores, human annotation of citation relevance, and numeric verification against ground truth—all future work.

Table 5. PROVENANCE ANALYSIS: RAG-ENABLED RESPONSES.

Metric	RAG Pipeline
Queries Tested	3
Average Latency	13,530 ms
Average Sources Cited	4.0 per response
Source Type	SEC 10-K filing excerpts
Cites Specific Fiscal Years	Yes (2022-2024)

Each RAG response cites an average of 4 text chunks from SEC 10-K filings, providing traceable references for qualitative claims. Without RAG, responses would rely solely on GPT-4o’s parametric knowledge without document grounding. We emphasize this demonstrates provenance capability, not accuracy validation.

C. Cost Analysis

Table 6 presents estimated API costs based on OpenAI pricing (December 2025). Estimates assume approximately 3,000 input tokens and 500 output tokens per query.

Table 6. API COST ESTIMATES (DECEMBER 2025 PRICING).

Usage	Est. Cost (USD)
Per Query (~3.5K tokens)	~\$0.0125
Per 100 Queries	~\$1.25
Per 1,000 Queries	~\$12.50

Based on GPT-4o at \$0.0025/1K input, \$0.01/1K output tokens. Embedding costs negligible (~\$0.000002/query).

VI. Limitations and Future Work

We explicitly acknowledge substantial limitations of this work:

Evaluation Limitations: This paper presents system-level metrics without rigorous benchmark evaluation. We do not evaluate on FinQA, ConvFinQA, FinanceBench, FinAR-Bench, or other established financial NLP datasets. We do not report retrieval quality metrics (NDCG@k, Recall@k) or faithfulness scores (RAGAS). The “sources cited” metric measures provenance, not correctness. Human evaluation of output quality is not conducted. These constitute critical gaps.

Structured Data Extraction: We do not parse XBRL or implement robust table extraction from SEC filings. KPIs are computed from SEC's JSON API structured fields stored in PostgreSQL, which may not capture all nuances (non-GAAP adjustments, restatements, amended filings). The RAG pipeline operates on narrative text only and cannot reliably extract numeric values from tables or charts.

Technical Design Choices: Key design decisions (L2 distance vs. cosine, chunk size selection, k=5 retrieval) lack empirical justification through ablation. We use ChromaDB's defaults without systematic comparison. Multi-agent prompts and orchestration logic are not published in detail.

Scope and Accuracy: Generated reports may contain hallucinations, numeric errors, or misleading analysis. The system should not be used for investment decisions without qualified human review. Edge cases (amendments, restatements, complex multi-year reconciliations) are not systematically evaluated.

Future Work: Priority future directions include: (1) evaluation on FinAR-Bench and ConvFinQA with strict numeric tolerances, (2) XBRL parsing for structured extraction, (3) RAGAS faithfulness evaluation, (4) human annotation studies with inter-rater reliability, (5) comparison against domain-specialized models (FinTral, Open-FinLLMs), (6) empirical comparison of distance metrics and retrieval configurations, and (7) detailed agent prompt and failure documentation.

VII. Conclusion

This paper presented GenAI Financial Reporter, a multimodal AI system for automated financial report generation. The system combines RAG with ChromaDB (1.3ms vector search), multi-agent orchestration, and integration with SEC EDGAR and Yahoo Finance. We demonstrate production deployment and open-source the implementation.

We emphasize this is primarily a systems paper describing practical deployment architecture. The evaluation demonstrates system functionality and provenance capability (4 sources cited per response), but does not validate accuracy, faithfulness, or benchmark performance. Rigorous evaluation against financial NLP benchmarks, structured XBRL extraction, and human annotation studies remain essential future work before any high-stakes application.

Supplementary Materials: Code is available at <https://github.com/Finance-Report-Generative-AI/Finance-Report-Generative-AI>.

Acknowledgments: The authors thank the Department of Computer Science at The Catholic University of America for supporting this work through the Artificial Intelligence course.

References

1. M. Kearney and S. Liu, "Textual sentiment in finance: A survey of methods and models," *Int. Review of Financial Analysis*, vol. 33, pp. 171-185, 2014.
2. A. El-Haj et al., "Multilingual financial text summarization," *Proc. 1st Financial Narrative Processing Workshop*, 2018.
3. D. Araci, "FinBERT: Financial sentiment analysis with pre-trained language models," *arXiv:1908.10063*, 2019.
4. H. Yang et al., "FinGPT: Open-source financial large language models," *arXiv:2306.06031*, 2023.
5. S. Wu et al., "BloombergGPT: A large language model for finance," *arXiv:2303.17564*, 2023.
6. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," *NeurIPS*, vol. 33, 2020.
7. OpenAI, "GPT-4 technical report," *arXiv:2303.08774*, 2023.
8. H. Chase, "LangChain," *github.com/langchain-ai/langchain*, 2022.
9. Financial Management Association, *Analyst Workflow and Reporting Practices*, 2023.
10. R. Bhatia et al., "FinTral: A Family of GPT-4 Level Multimodal Financial LLMs," *arXiv:2402.10986*, 2024.
11. Q. Xie et al., "Open-FinLLMs: Open Multimodal LLMs for Financial Applications," *arXiv:2408.11878*, 2024.

12. J. Chen et al., "MultiFinRAG," *arXiv:2506.20821*, 2025.
13. A. Kumar et al., "MarketSenseAI 2.0," *arXiv:2502.00415*, 2025.
14. Z. Zhang et al., "FinAR-Bench: Financial Analysis and Reasoning Benchmark," *arXiv:2506.07315*, 2025.
15. Y. Wang et al., "REAL-MM-RAG: Real-world Multi-Modal RAG," *arXiv:2502.12342*, 2025.
16. L. Li et al., "MDocAgent: Multi-Document Agent," *arXiv:2503.13964*, 2025.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.