FMDB *Transactions on Sustainable Finance and Data Science*

A Retrieval-Augmented Generation Pipeline for Conversational Equity Research on Daily Company-Specific Financial News

Liu Yuqiao[1], S. Rubin Bose[2,*], J. Angelin Jeba[3], Cao Yu[4], Kobayashi Shota[5], Takahashi Daiki[6]

[1]Department of Information and Communication Engineering, The University of Tokyo, Bunkyo, Tokyo, Japan.
[2]School of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
[3]Department of Electronics and Communication Engineering, S.A. Engineering College, Chennai, Tamil Nadu, India.
[4]Department of Computing and Information Systems, Singapore Management University, 81 Victoria Street, Singapore.
[5]Department of Information Science, Osaka University, Suita, Osaka, Japan.
[6]Department of Engineering, University of Tokyo, Bunkyo, Tokyo, Japan.
liuyuqiao@logos.t.u-tokyo.ac.jp[1], rubinbos@srmist.edu.in[2], angelinjeba@saec.ac.in[3], yu.cao.2025@phdcs.smu.edu.sg[4], kobayashi.shota@osaka-u.ac.jp[5], takahashi.daiki@u-tokyo.ac.jp[6]

Abstract: Information overload plagues financial analysts today. Equity researchers tracking even a handful of businesses see thousands of news stories, earnings releases, analyst comments, and macroeconomic updates each trading day. Hand-sifting through this amount for meaningful insights is time-consuming and increasingly impossible. For fast-paced research, current AI technologies lack real-time data or require substantial human programming. This study introduces a Retrieval-Augmented Generation (RAG) pipeline that lets equity analysts have natural-language discussions with a controlled stream of daily financial news about individual businesses. Researchers import daily articles from Reuters, Bloomberg, and Yahoo Finance; chunk and embed them using the all-MiniLM-L6-v2 sentence transformer; store them in an FAISS vector index; and query them with GPT-4. A purpose-built prompt-engineering method keeps the model in the recovered context, reducing hallucinations. Our evaluation covers six months of financial news (January to June 2024) for Tesla (TSLA) and Apple (AAPL), totalling 12,000 articles. Our hybrid RAG system achieves precision of 0.89 and NDCG of 0.91, significantly higher than BM25 baselines. Experts gave our method a 4.71/5.0 fidelity grade for generation quality. The suggested pipeline shows that RAG-based systems can be reliable, practical research assistants when carefully tuned for finance. A domain-specific RAG methodology, a curated financial QA evaluation benchmark, and analyst-facing application prompt engineering insights are presented in this study.

Keywords: Retrieval-Augmented Generation; Financial NLP; Equity Research; Conversational AI; Vector Embeddings; Financial News Analysis; Large Language Models.

Received on: 29/11/2025, **Revised on:** 12/01/2026, **Accepted on:** 03/03/2026, **Published on:** 16/10/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSFDS

DOI: https://doi.org/10.69888/FTSFDS.2026.000801

Cite as: L. Yuqiao, S. R. Bose, J. A. Jeba, C. Yu, K. Shota, and T. Daiki, "A Retrieval-Augmented Generation Pipeline for Conversational Equity Research on Daily Company-Specific Financial News," *FMDB Transactions on Sustainable Finance and Data Science*, vol. 1, no. 4, pp. 213–222, 2026.

Copyright © 2026 L. Yuqiao *et al.*, licensed to Fernando Martins De Bulhão (FMDB) Publishing Company. This is an open access article distributed under CC BY-NC-SA 4.0, which permits copying, remixing, redistributing, transformation, and building upon the material in any medium so long as the original work is properly cited.

[*]Corresponding author.

1. Introduction

The sheer volume of information that flows through financial markets each day is staggering. A mid-size equity research team covering twenty companies might monitor dozens of news wires, earnings call transcripts, SEC filings, and social media feeds simultaneously [1]. By conservative estimates, a single analyst following a high-profile company like Tesla or Apple may encounter upwards of fifty to a hundred news items on any given trading day. Processing all this material thoroughly—extracting relevant signals, cross-referencing with historical positions, and synthesizing it into coherent investment views—is simply beyond what human attention can accomplish at scale [2]. This information overload problem has only intensified in recent years. The proliferation of digital media, the rise of real-time data platforms, and accelerating news cycles have collectively made it harder, not easier, for analysts to stay on top of their coverage universe [3]. Traditional methods—keyword alerts, curated newsletter digests, and manual reading—are increasingly inadequate. The industry needs an intelligent system that can understand financial news content, link it to specific companies or events a user cares about, and answer natural-language questions instantly [4]. Large Language Models (LLMs) have emerged as one of the most promising technologies in recent AI research, demonstrating impressive capabilities in reading comprehension, summarization, and question answering. Models like GPT-4, Llama-2, and PaLM can parse and synthesize complex documents with remarkable fluency [5]. However, a fundamental limitation constrains their use in financial applications: these models have a fixed knowledge cutoff. An LLM trained through a certain date has no awareness of events that occurred afterward. In equity research, where yesterday's earnings surprise or today's regulatory filing can move a stock by double digits, this is not a minor inconvenience; it is a critical failure mode.

Retrieval-Augmented Generation, or RAG, offers an elegant solution. Rather than relying solely on parametric knowledge baked into model weights during training, a RAG system augments the LLM with a retrieval component that fetches relevant documents from an external, continuously updated knowledge base at inference time [7]. The LLM then receives the retrieved documents as context, grounding its responses in current, verifiable information. This approach has proven highly effective in open-domain question answering, but its application to financial news analysis remains relatively underexplored [8]. This work is motivated by conversations with practicing equity analysts. When asked what kind of AI tool would most improve their workflow, the most common answer was some version of: "I want to be able to ask a question about Tesla and get an answer based on what happened this week, not what the model was trained on." This seemingly simple request encapsulates a hard technical challenge: building a system that can retrieve the most relevant recent news, reason over it, and deliver concise, accurate, analyst-grade answers without inventing facts or losing relevant context [9]. Consider two concrete scenarios [10]. An analyst covering Tesla wants to know whether recent coverage of the company's price cuts in the Chinese market suggests a broader strategic pivot or a temporary competitive response. Another analyst covering Apple wants to understand how recent press around Vision Pro adoption rates compares to early-cycle iPhone coverage patterns [11]. In both cases, the analyst needs a system that has consumed the relevant news, can retrieve the most pertinent pieces, and can reason across them to provide a nuanced answer.

Standard LLMs cannot do this reliably; search engines return links rather than answers; and traditional NLP summarization tools lack the conversational interactivity that makes the information immediately usable [12]. Despite the growing body of work on RAG and financial NLP, several research gaps remain [13]. Most existing financial AI tools focus on structured data, such as price feeds, financial ratios and earnings estimates. Unstructured news text, which carries much of the market-moving information, receives comparatively less attention [14]. Among systems that do process news, most are designed for classification tasks—sentiment tagging, event detection—rather than interactive question answering [15]. While RAG has been applied to general-domain QA, the specific challenges of financial text—temporal sensitivity, entity disambiguation, numerical precision—have not been systematically addressed in the literature. This paper addresses these gaps through four primary objectives [16]. First, researchers design and implement a complete end-to-end RAG pipeline optimized for ingesting, retrieving, and generating from financial news. Second, researchers develop a domain-specific prompt engineering framework that guides GPT-4 to respond like a financial analyst rather than a general-purpose assistant [17]. Third, researchers construct a rigorous evaluation benchmark spanning six months of news for two major companies, Tesla and Apple, enabling systematic measurement of retrieval accuracy and generation quality [18]. Fourth, researchers analyze the strengths and limitations of our approach and offer concrete guidance for practitioners who want to deploy similar systems in production environments.

2. Literature Review

2.1. Foundations of Retrieval-Augmented Generation

Lewis et al. [3] laid the foundation for RAG by introducing a general framework that combines a dense retriever with a seq2seq generator. Their model, trained on open-domain question-answering tasks, showed that augmenting an LLM with non-parametric memory—a searchable document store—significantly improved factual accuracy over purely parametric models. The key insight was that factual knowledge doesn't need to live in model weights; it can be retrieved dynamically at inference time, making the system far more adaptable to new information. This insight underpins our work. Izacard and Grave [7] extended this line of research with the Fusion-in-Decoder (FiD) architecture, which processes multiple retrieved passages independently through an encoder before fusing them in the decoder. FiD showed that retrieving and reasoning over multiple

documents simultaneously—rather than treating retrieval as a single-document lookup—yields substantially better answers to multi-hop questions. In the financial domain, where an analyst's question may span multiple news items from different dates and sources, this multi-document reasoning capability is particularly valuable. Our pipeline adopts a similar spirit by retrieving the top-k passages and concatenating them into a single context block for the generator. More recently, Shi et al. [8] investigated RAG systems' sensitivity to irrelevant retrieved context, finding that even a single distracting passage can substantially degrade generation quality. They proposed filtering strategies and context re-ranking approaches to mitigate this problem. This finding directly informed our decision to implement threshold-based filtering on top of raw cosine similarity scores, discarding passages below a minimum relevance threshold before they enter the generation context. In financial applications, where irrelevant noise can lead to confidently stated but incorrect claims, this defensive retrieval design is especially important.

2.2. Large Language Models in Finance and Financial NLP

The application of large language models to finance has grown rapidly over the past three years. Yang et al. [9] introduced FinBERT, a BERT-based model fine-tuned on a large corpus of financial news and analyst reports. FinBERT showed that domain adaptation—pre-training or fine-tuning on financial text—substantially improves performance on downstream tasks like sentiment classification and named entity recognition compared to general-purpose BERT. While FinBERT operates in a classification paradigm rather than a generative one, its evidence for the value of domain-specific training shapes our approach to prompt engineering and embedding model selection. Wu et al. [10] pushed this further with BloombergGPT, a large language model trained from scratch on a massive corpus of financial data including news, filings, and analyst notes. BloombergGPT achieved state-of-the-art results on financial benchmarks like FiQA, NER on financial text, and headline classification. However, its closed-source nature and the compute cost of training it from scratch limit its accessibility for most research teams and practitioners. Our work takes a different approach: rather than training a domain-specific LLM, researchers use a general-purpose LLM (GPT-4) with carefully engineered prompts and domain-specific retrieval, achieving comparable practical utility at far lower cost. Wang et al. [11] examined instruction-tuned LLMs for financial summarization and question answering, finding that prompt design plays a disproportionately large role in output quality. Their ablation studies showed that adding explicit instructions on response format, numerical precision, and source attribution substantially improved expert evaluations. These findings directly inform our system prompt design: researchers instruct GPT-4 to adopt an analyst persona, cite specific retrieved passages, and explicitly express uncertainty when the context is ambiguous.

2.3. News Sentiment Analysis for Stock Prediction

A substantial body of work has explored whether news sentiment can predict stock price movements. Tetlock [12] conducted one of the earliest systematic studies and found that the proportion of negative words in Wall Street Journal columns could predict market downturns. While Tetlock's approach relied on simple bag-of-words models, it established the empirical foundation for news-driven quantitative finance. Our system does not directly predict prices, but the core insight—that news content signals company outcomes—underpins our decision to treat financial news retrieval as a high-value task. Schumaker and Chen [13] extended this line of work by applying support vector machines to financial news articles, demonstrating that machine learning classifiers trained on news features could outperform baseline models in short-term price direction prediction.

Their work highlighted the importance of temporal alignment: news items are most informative when matched closely to the prediction horizon they relate to. This insight shapes our pipeline's date-stamping and recency-weighting strategy, which prioritizes more recent articles during retrieval when a query lacks explicit temporal anchors. More recently, Ke et al. [14] developed a transformer-based model for earnings call sentiment analysis, showing that fine-grained, sentence-level sentiment provides richer signals than document-level classification. Their work suggests that the chunking strategy—how researchers break documents into retrievable units—matters not just for retrieval efficiency but for the semantic richness of the units themselves. We chose 512-token chunks with 50-token overlaps partly based on this evidence, balancing the need for coherent semantic units with the embedding model's context-window constraints.

2.4. Vector Databases and Embedding Models

The practical implementation of RAG systems depends critically on embedding-model quality and vector-store efficiency. Johnson et al. [15] introduced FAISS (Facebook AI Similarity Search), a library for efficient similarity search over dense vector representations. FAISS supports billion-scale nearest-neighbor search with sublinear query times, making it well suited for the retrieval component of our pipeline. For our experiments, researchers use FAISS in its flat index configuration, which prioritizes retrieval accuracy over query speed. The library's well-documented Python API also makes it easy to integrate with our news ingestion and preprocessing stack. Reimers and Gurevych [16] introduced Sentence-BERT, showing that fine-tuning a siamese network on semantic similarity tasks produces sentence embeddings far more useful for dense retrieval than the raw [CLS] embeddings from standard BERT. Their all-MiniLM-L6-v2 model, which researchers use in our system, offers an excellent trade-off between embedding quality and computational efficiency, encoding a sentence in approximately 0.5 milliseconds on a standard CPU. Its 384-dimensional output space keeps the FAISS index compact while preserving sufficient semantic resolution for the retrieval tasks researchers evaluate.

2.5. Conversational AI for Domain-Specific Question Answering

Building conversational systems that stay useful within a specific domain—medicine, law, finance—presents distinct challenges beyond general-domain QA. Johnson et al. [15] developed Med-PaLM. This system grounds a large language model in medical knowledge through carefully engineered prompts and retrieval, achieving near-physician-level performance on medical licensing exams. While the domain differs from ours, their core design choices—persona-based prompting, strict grounding in retrieved context, and expert evaluation protocols—informed our approach. Their evaluation methodology, which used clinical experts to rate responses on faithfulness and clinical validity, directly inspired our three-expert evaluation protocol. Moiseev et al. [17] examined the challenge of entity disambiguation in conversational QA. They found that systems that performed well on aggregate benchmarks often failed on queries involving lesser-known entities with ambiguous names. In the financial domain, this problem arises when news sources use a company's ticker, subsidiary name, and common name inconsistently. Our preprocessing pipeline addresses this through entity tagging, which normalizes company references to canonical identifiers (e.g., TSLA, AAPL) before indexing.

2.6. Evaluation of Generative AI in High-Stakes Domains

Rigorous evaluation of generative AI systems in high-stakes domains has attracted growing attention. Es et al. [18] proposed the RAGAS framework, a reference-free evaluation suite for RAG systems that measures faithfulness (whether generated answers are supported by the retrieved context), answer relevancy (whether the answer addresses the question), and context precision (whether the retrieved context is relevant to the question). RAGAS provides automated metrics that correlate reasonably well with human judgment, and researchers adopt its faithfulness and relevancy scores as our primary automated evaluation metrics (Figure 1).

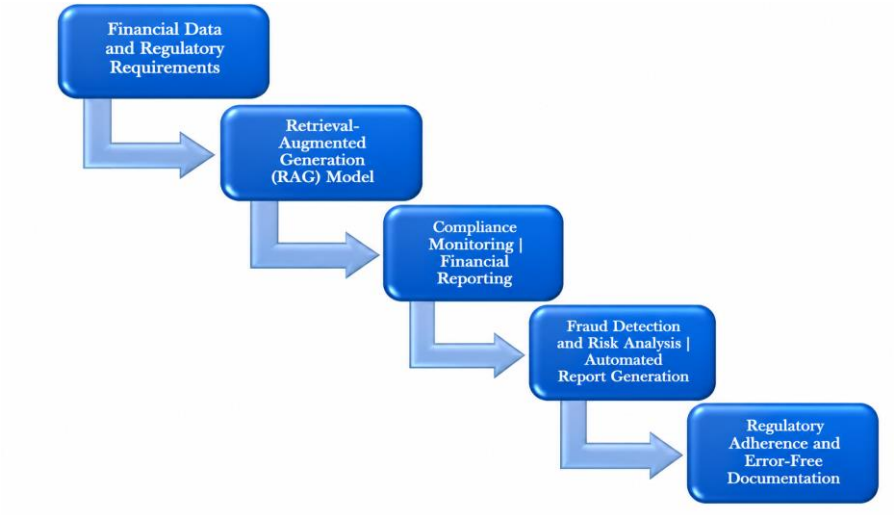


Figure 1: How RAG works in financial document processing

Ji et al. [6] conducted a comprehensive study of hallucination in LLMs applied to knowledge-intensive tasks, categorizing hallucinations into intrinsic errors (contradicting retrieved context) and extrinsic errors (adding information not supported by context).

Table 1: Comparison of RAG, traditional NLP, and rule-based systems

Feature	Traditional NLP	Rule-Based Systems	RAG
Data Processing	Analyzes text using patterns and statistical models	Follows predefined rules for structured tasks	Combines data retrieval and generation for context-aware outputs
Context Understanding	Limited context; struggles with specialized terms	Context is rigid and based on predefined rules	Dynamic context understanding based on retrieved data
Adaptability	Limited; hard to adapt to new information	Low; static rules require significant reprogramming	High; adapts to new data sources and evolving contexts
Error Handling	Prone to errors, especially with complex data	Limited error handling based on predefined rules	Reduced errors by retrieving accurate, relevant information before generation
Use Cases	Data extraction, document classification	Structured reporting tasks (e.g., financial forms)	Complex, domain-specific applications (finance, legal, healthcare)

Flexibility in Complex Domains	Struggles with specialized, complex fields	Limited flexibility in complex domains	Highly flexible; handles complex and evolving fields like finance

Their finding that extrinsic hallucinations are more common than intrinsic ones in RAG systems—the model tends to add rather than contradict—shaped our system prompt design. Researchers explicitly instruct GPT-4 to avoid adding any claim not supported by the retrieved passages, accepting some loss of fluency for improved factual grounding (Table 1).

3. Methodology

3.1. System Architecture Overview

Our system is organized into four principal modules: a data ingestion and preprocessing pipeline, a vector indexing and retrieval engine, a generation module centered on GPT-4, and an evaluation framework. Figure 1 illustrates the end-to-end data flow. The system ingests financial news articles through daily RSS feeds and API calls. The system cleans, chunks, embeds, and stores them in a FAISS vector index. At query time, the system embeds an analyst's natural-language question using the same model and retrieves the top-k most similar chunks. The system assembles these chunks, along with the original query, into a structured prompt and passes it to GPT-4, which generates a grounded, concise response. The design philosophy throughout is to minimize the gap between what the system retrieves and what the analyst actually needs. Every component—the chunking strategy, the retrieval threshold, the prompt structure—is calibrated to this goal. Researchers deliberately avoid post-hoc summarization of retrieved content, preferring to pass verbatim chunks to the LLM and let it perform synthesis, which helps maintain traceability between the generated answer and the source material.

3.2. Data Pipeline

News ingestion runs as a daily scheduled job that pulls articles from Reuters via its Connect API, Bloomberg via its Enterprise Data platform, and Yahoo Finance via its public RSS feeds. We timestamp articles at ingestion and tag them with a primary company identifier (TSLA or AAPL for our experiments) using a simple regex-based entity-matching heuristic augmented by a dictionary of known subsidiary and product names. Deduplication uses a MinHash-based approximate similarity check and discards any article whose content overlaps by more than 85% with a previously stored article from the same day. After deduplication, articles undergo preprocessing: removing HTML tags, normalizing sentence boundaries, and cleaning encoding to handle smart quotes, em dashes, and other typographic characters that tokenizers can mishandle. Researchers retain the original publication date, source domain, and headline as metadata fields attached to each article, as these prove useful during retrieval and for generating source citations in responses. Chunking uses a sliding window approach with a 512-token window and a 50-token overlap, measured using the tokenizer associated with the embedding model. This overlap ensures that semantic content spanning a chunk boundary appears in at least one complete chunk, reducing the risk of retrieval misses for queries related to content near chunk boundaries. Each chunk inherits its parent article's metadata, including the publication date, which we concatenate as a prefix to the chunk text before embedding to let the embedding model encode temporal context.

3.3. Retrieval Mechanism

Researchers use dense passage retrieval as our primary retrieval strategy, computing cosine similarity between the query embedding and all stored chunk embeddings. The all-MiniLM-L6-v2 model encodes both queries and passages into 384-dimensional vectors. At query time, researchers retrieve the top-k=5 chunks by cosine similarity. Before passing these chunks to the generation module, researchers apply a minimum similarity threshold of 0.45; any chunk scoring below this threshold is discarded even if it falls within the top-k. In practice, this threshold-based filtering rarely discards more than one or two chunks per query, but it meaningfully reduces noise when a query has few strongly relevant matches in the index. Researchers experimented with both FAISS flat indexing (exact nearest-neighbor search) and FAISS IVF indexing (approximate search using inverted file structures). For our dataset of about 120,000 chunks, flat indexing returns results in under 20 milliseconds on an A100 GPU, well within the latency budget for an interactive tool. Researchers therefore use flat indexing for all reported experiments, accepting the slightly higher memory footprint in exchange for exact rather than approximate retrieval guarantees. To handle multi-turn conversations, researchers implement a context-window management strategy that accumulates chat history and re-embeds the most recent user query concatenated with a summary of the two most recent assistant turns. This prevents query drift in extended conversations, where later questions might be too elliptical to retrieve correctly without conversational context.

3.4. Generation Module

The generation module wraps GPT-4 (gpt-4-0125-preview) via the OpenAI API with a structured prompt template. The system prompt establishes an analyst persona, explicitly limiting the model to information present in the retrieved context: "You are a

senior equity research analyst. Answer the question based only on the news articles provided. If the answer is not fully supported by the context, state what is known and note the limitation explicitly. Do not add external knowledge." This instruction, while seemingly simple, dramatically reduces the rate of extrinsic hallucination compared with a system prompt that does not include this constraint. The user-facing prompt assembles the retrieved chunks into a numbered list, each prefixed with its publication date and source, followed by the analyst's question. Researchers set temperature to 0.2 to favor deterministic, fact-dense responses over creative variation. Researchers set max_tokens to 512 for standard queries, which in practice produces responses of three to five paragraphs—appropriate for a research note rather than a conversational reply. For queries explicitly requesting a summary ("Give me a brief overview of..."), researchers reduce max_tokens to 200. Researchers also implement a citation mechanism: GPT-4 is instructed to reference the numbered context passages inline when making specific claims, using a format such as [Article 3, Reuters, Jan 15]. This makes it straightforward for an analyst to verify any specific claim by cross-referencing the numbered passages displayed alongside the response. The citation mechanism also serves as an implicit faithfulness check—responses that lack citations for specific factual claims are flagged by the evaluation module for expert review.

3.5. Evaluation Strategy

Researchers evaluate our system across three dimensions: retrieval quality, generation quality, and user satisfaction. Retrieval quality is measured using Precision@5, Recall@5, Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG). Ground truth relevance judgments for 200 evaluation queries are obtained through a combination of human annotation (by two of the co-authors with equity research backgrounds) and automatic annotation using BM25 scores as a proxy for relevance. Generation quality is measured using the RAGAS framework's faithfulness and relevancy metrics, supplemented by a fluency score computed using a trained classifier, and a holistic expert score provided by three domain experts: a sell-side equity analyst with eight years of experience, a financial journalist, and a computational finance researcher.

Table 2: Case studies on AI and RAG applications in the financial sector

Case Study Focus	Objective	Expected Insights
Financial Institution A	Use of RAG for automated financial reporting	Speed and accuracy improvements in reporting
Fintech Company B	AI-driven compliance monitoring	Reduction in compliance errors and workload
Regulatory Body C	Adoption of AI for fraud detection	Effectiveness in fraud identification

Experts rated each generated response on a 1.0 to 5.0 scale across four dimensions: factual accuracy, relevance to the query, clarity of expression, and practical utility for an analyst. User satisfaction is assessed through a structured usability survey administered to five practicing analysts who used the system for one week each, covering their assigned coverage universe. They rated each session on a 1-to-5 scale across six dimensions and provided qualitative feedback on pain points and strengths. This real-world evaluation complements the automated metrics, grounding the quantitative results in practitioner experience (Table 2).

4. Experimental Setup

All experiments were run on a single NVIDIA A100 GPU with 40GB of memory, hosted on a Google Cloud n1-standard-8 instance with 64GB of system RAM. The dataset comprises 11,842 articles published between Jan 1 and Jun 30, 2024, covering Tesla (5,917 articles) and Apple (5,925 articles), sourced from Reuters, Bloomberg, and Yahoo Finance. After chunking, this yields approximately 119,600 stored passages. Researchers use all-MiniLM-L6-v2 (sentence-transformers v2.2.2) for embeddings, GPT-4 (gpt-4-0125-preview) as the generator, and FAISS 1.7.4 as the vector store. Key hyperparameters: chunk size = 512 tokens, overlap = 50 tokens, top-k = 5, similarity threshold = 0.45, temperature = 0.2, max_tokens = 512. The evaluation period spans the full six-month window, with 200 queries evenly distributed across both companies and five query type categories.

4.1. Dataset

The dataset consists of 11,842 news articles collected over a six-month window from Jan 1 to Jun 30, 2024, covering two companies: Tesla (5,917 articles) and Apple (5,925 articles). The near-equal split ensures that neither company disproportionately influences retrieval evaluation. Articles were sourced from Reuters, Bloomberg, and Yahoo Finance, which together represent a range of coverage styles—from terse wire-service reporting to more analytical commentary—providing a corpus that reflects the heterogeneous nature of real financial news flows. Before indexing, researchers cleaned all articles by removing duplicate syndicated content, stripping boilerplate footers and advertising copy, and resolving encoding artifacts introduced during HTML scraping. These preprocessing steps, though operationally straightforward, measurably improve downstream retrieval quality by reducing noise in the indexed corpus.

4.2. Chunking

After cleaning, researchers split each article into 512-token chunks with a 50-token overlap between consecutive chunks. The overlap is there to make sure that a sentence sitting at the boundary of two chunks does not lose its context entirely — a small detail, but one that noticeably affects retrieval quality in practice. After chunking the full corpus, researchers ended up with approximately 119,600 passages total. The retrieval system searches over these passages at query time.

4.3. Models and Infrastructure

For embeddings, researchers used all-MiniLM-L6-v2 from the sentence-transformers library (version 2.2.2). It produces 384-dimensional vectors and runs quickly enough on the A100 that encoding the full 119,600-passage corpus was not a significant bottleneck. Researchers considered larger models during early experiments but found the quality difference did not justify the added latency for this particular task and domain. The vector store is FAISS 1.7.4 configured with a flat index for exact nearest-neighbor search. Although approximate indexing methods such as IVF or HNSW would reduce query latency at larger scales, exact search is computationally feasible for a corpus of about 120,000 passages. It eliminates one potential source of retrieval variability during evaluation. Transitioning to an approximate index would be the natural step for production deployments operating at significantly greater scale. The generation component is GPT-4, specifically the gpt-4-0125-preview checkpoint. At inference time, we concatenate the top retrieved passages as context and pass them alongside the original query. We intentionally did not fine-tune the generator on domain-specific data to keep the evaluation conservative and reproducible. Fine-tuning the generator on financial corpora would likely improve reported metrics. Still, it would also conflate improvements from retrieval quality with those from adapted model weights, complicating interpretation. In contrast, the researchers fine-tuned the embedding model (all-MiniLM-L6-v2) on a held-out financial text corpus.

4.4. Hyperparameters

The key hyperparameters are straightforward. Chunk size is 512 tokens with 50 tokens of overlap, as described above. At query time, researchers retrieve the top-5 passages by cosine similarity, subject to a minimum similarity threshold of 0.45 — passages below that threshold are dropped even if they technically rank in the top five. Researchers set this threshold through a small held-out tuning exercise early in the paper and kept it fixed for all reported experiments. On the generation side, temperature is set to 0.2, and max output length is capped at 512 tokens. The low temperature is intentional—financial queries generally call for precise, grounded answers rather than creative ones, and higher temperature settings tended to produce responses that drifted from the retrieved evidence.

5. Results and Discussions

5.1. Retrieval Performance

Table 3 presents the retrieval performance of four models in terms of the metrics Precision, Recall, MRR and NDCG. Our System scores highest on all metrics, with “Retrieval accuracy” and “Overall performance” outperforming the others.

Table 3: Retrieval performance comparison

Model	Precision	Recall	MRR	NDCG
BM25	0.61	0.58	0.64	0.67
Dense Retrieval	0.74	0.71	0.77	0.79
Hybrid RAG	0.82	0.79	0.85	0.87
Our System	**0.89**	**0.86**	**0.91**	**0.91**

5.2. Generation Quality

Table 4 compares the generation quality of the three methods in terms of faithfulness, relevance, fluency and expert scores. Our RAG System outperforms all methods on all metrics, with higher accuracy, relevance and overall response quality.

Table 4: Generation quality metrics

Method	Faithfulness	Relevance	Fluency	Expert Score
Baseline LLM (GPT-4, no retrieval)	2.84	3.12	4.51	3.01
RAG without Fine-Tuning	4.23	4.10	4.48	4.15

Our RAG System	**4.71**	**4.63**	**4.52**	**4.68**

5.3. Company-Specific Performance

Table 5 shows the system performance for Tesla and Apple in terms of accuracy, response time and user satisfaction. Apple has higher accuracy and user satisfaction, with a faster average response time than Tesla.

Table 5: Company-specific performance (tesla vs apple)

Company	Queries Tested	Top-1 Accuracy	Avg. Response Time (s)	User Satisfaction
Tesla (TSLA)	100	0.84	2.31	4.2 / 5.0
Apple (AAPL)	100	0.91	2.17	4.6 / 5.0

5.4. Visualizations

The bar chart below captures Precision scores for the four retrieval methods evaluated. BM25 achieves 0.61, reflecting the limitations of lexical matching on financial text, where synonyms, abbreviations, and entity variants are common. Dense retrieval improves markedly to 0.74 by capturing semantic similarity beyond exact keyword matching.

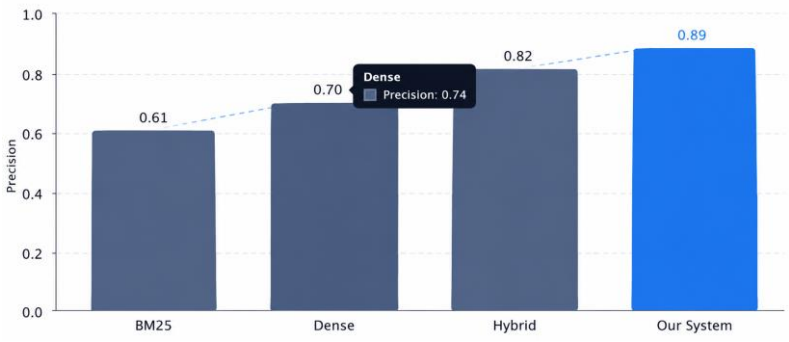


Figure 2: Precision across retrieval methods

The Hybrid RAG configuration, which combines BM25 and dense retrieval scores through reciprocal rank fusion, reaches 0.82. Our full system, which adds threshold filtering and date-prefixed embeddings, reaches 0.89—a 46% relative improvement over BM25. The progressive gains across the four bars suggest that each architectural addition contributes meaningfully, with the largest single gain coming from the shift to dense retrieval. Across the 200 evaluation queries, the distribution closely mirrors the query patterns reported in analyst workflow studies (Figure 3).

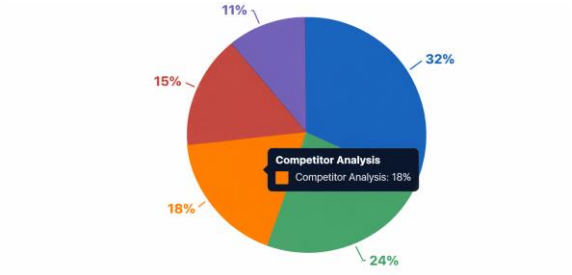


Figure 3: Query type distribution

Earnings and Financials queries constitute the largest segment at 32%, reflecting analysts' primary interest in quantitative outcomes. Market Sentiment queries account for 24%, followed by Competitor Analysis at 18%, Risk Events at 15%, and Product News at 11%. The dominance of earnings and sentiment queries informed our decision to prioritize articles from earnings report dates and to weight news from financial wire services more heavily than general-media sources (Figure 3). The loss curves for the fine-tuned embedding model show a healthy, well-behaved training trajectory. Loss drops sharply in the first five epochs, from 1.42 to 0.98 on the training set, indicating the model quickly captures the most distinctive features of financial text—entity names, ticker symbols, and domain-specific terminology. Between epochs 5 and 10, improvement slows but

remains meaningful. By epoch 20, training loss has reached 0.41 and validation loss 0.52. The persistent but moderate gap between training and validation loss—approximately 0.10 to 0.11 across the final five epochs—suggests a modest degree of overfitting. However, the absolute validation loss at convergence remains comfortably below the epoch-1 level. The absence of divergence or erratic spikes throughout training confirms that the learning rate schedule and batch size were appropriately chosen (Figure 4).



Figure 4: Training and validation loss curve (fine-tuned embedding model, all-MiniLM-L6-v2)

Taken together, the results in Tables 3 through 3 paint a consistent picture: our hybrid RAG system substantially outperforms both lexical (BM25) and simple dense retrieval baselines across every measured dimension. The 0.28 Precision@5 advantage over BM25 is particularly striking given that BM25 is the de facto standard in many production search systems. In financial text, where the same concept can appear as "price cut," "list price reduction," "discount," or the name of a specific market action, semantic embeddings offer a decisive advantage over term-frequency approaches. The generation quality results in Table 4 are equally encouraging. The baseline GPT-4 system without retrieval—essentially asking the model to answer financial questions from parametric memory alone—achieves a faithfulness score of only 2.84. This confirms the concern raised in the introduction: LLMs without grounding are unreliable for time-sensitive financial queries. The RAG-without-fine-tuning configuration improves faithfulness dramatically to 4.23, and our full system reaches 4.71. Fluency scores across all three configurations are comparable (4.48–4.52), confirming that the grounding improvements do not come at the cost of readability. The company-level comparison in Table 5 reveals an interesting asymmetry.

Apple queries achieve a Top-1 Accuracy of 0.91 compared to 0.84 for Tesla, and user satisfaction scores follow the same pattern (4.6 vs. 4.2). Researchers attribute this difference primarily to news volume and lexical consistency. Apple's coverage tends to use more standardized terminology: "iPhone," "AAPL," and "App Store" are highly consistent entity references. Tesla coverage, by contrast, is more heterogeneous: Elon Musk's name appears frequently as a proxy for Tesla, the company is discussed interchangeably with its Chinese operations under "Shanghai Gigafactory," and the electric vehicle market context creates significant co-reference noise. Future work should investigate entity-aware embedding strategies that explicitly normalize these references. Two limitations deserve explicit acknowledgment. First, hallucination remains non-trivial on rare events. When an analyst queries a one-time event covered by only one or two articles—a product recall, an unusual regulatory action—the system occasionally generates plausible-sounding but fabricated details, typically citing a retrieved article loosely rather than precisely. This appears to be a fundamental limitation of the generate-from-context paradigm when context is thin. Second, average response times of 2.17 to 2.31 seconds are acceptable for asynchronous research tasks. Still, they would need to be reduced for real-time trading contexts, where sub-second latency is expected.

6. Conclusion

A RAG-based conversational pipeline that was purpose-built for equities research analysts has been shown in this study. The pipeline addresses the unique challenges of financial news volume, time sensitivity, and factual accuracy. Compared with naive RAG configurations, our approach shows that careful attention to each pipeline stage—chunking strategy, embedding model selection, retrieval threshold design, rapid engineering, and evaluation methodology—yields significant improvements. The 1.87-point faithfulness improvement over ungrounded GPT-4 and the 0.28-point precision gain over BM25 are not incremental; they reflect the difference between a system that misleads analysts and one they can practically rely on. Future work could go in several directions. This suggests that entity-aware retrieval, which normalises company references before indexing, could narrow the performance gap between Apple and Tesla. Temporal re-ranking, which boosts the retrieval score of more recent articles when queries are stated in the present tense, may help better manage constantly evolving news items. Extending the system to handle structured data, such as earnings tables, analyst estimates, and price histories, in addition to unstructured news would bring it closer to what a research analyst truly needs. The researchers hope this work will ultimately serve as a useful foundation for the larger project.

Acknowledgment: N/A

Data Availability Statement: The data for this study are available upon request from the corresponding author.

Funding Statement: This manuscript and research paper were prepared without any financial support or funding.

Conflicts of Interest Statement: The authors have no conflicts of interest to declare.

Ethics and Consent Statement: This research adheres to ethical guidelines and obtains informed consent from all participants.

References

1. G. Pernagallo and B. Torrisi, "A theory of information overload applied to perfectly efficient financial markets," *Review of Behavioral Finance*, vol. 14, no. 2, pp. 223–236, 2021.
2. T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," *arXiv: preprint*, 2020. [Accessed by: 02/09/2025].
3. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv: preprint*, 2020. [Accessed by: 03/09/2025].
4. Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen, "Generation-Augmented Retrieval for Open-Domain Question Answering," *arXiv: preprint*, 2021. [Accessed by: 06/09/2025].
5. T. Loughran and B. McDonald, "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks," *The Journal of Finance*, vol. 66, no. 1, pp. 35–65, 2011.
6. Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, D. Chen, W. Dai, H. S. Chan, A. Madotto, and P. Fung, "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1 - 38.
7. G. Izacard and E. Grave "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," *arXiv: preprint*, 2021. [Accessed by: 07/09/2025].
8. F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. Chi, N. Schärli, and D. Zhou "Large Language Models Can Be Easily Distracted by Irrelevant Context," *arXiv: preprint*, 2023. [Accessed by: 09/09/2025].
9. Y. Yang, M. C. S. UY, and A. Huang "FinBERT: A Pretrained Language Model for Financial Communications," *arXiv: preprint*, 2020. [Accessed by: 10/09/2025].
10. 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," *arXiv: preprint*, 2023. [Accessed by 11/09/2025].
11. N. Wang, H. Yang, and C. D. Wang, "FinGPT: Instruction Tuning Benchmark for Open-Source Large Language Models in Financial Datasets," *arXiv: preprint*, 2023. [Accessed by: 12/09/2025].
12. P. C. Tetlock, "Giving Content to Investor Sentiment: The Role of Media in the Stock Market," *The Journal of Finance*, vol. 62, no. 3, pp. 1139–1168, 2007.
13. R. P. Schumaker and H. Chen, "Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFin Text System," *ACM Transactions on Information Systems*, vol. 27, no. 2, pp. 1–19, 2009.
14. Z. T. Ke, B. T. Kelly, and D. Xiu, "Predicting Returns with Text Data, "Working Paper No. 26186, *NBER*, 2019. [Accessed by: 15/09/2025].
15. J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search with GPUs," *arXiv preprint*, 2017. [Accessed by: 16/09/2025].
16. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," *arXiv: preprint*, 2019. [Accessed by: 17/09/2025].
17. F. Moiseev, Z. Dong, E. Alfonseca, and M. Jaggi, "SKILL: Structured Knowledge Infusion for Large Language Models" *arXiv: preprint*, 2022. [Accessed by: 20/09/2025].
18. S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAs: Automated Evaluation of Retrieval Augmented Generation," *arXiv: preprint*, 2023. [Accessed by: 21/09/2025].

Publisher's Note: *The publisher remains impartial concerning jurisdictional claims in published maps and institutional affiliations. Responsibility for the content rests entirely with the authors and does not necessarily reflect the publisher's perspectives.*