ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 5 Issue 4 October 2025 / Page No: 47-57
Paper Id: JETA-V5I4P108/ Doi: 10.56472/25832646/JETA-V5I4P108

Original Article

Beyond Static Retrieval: Evaluating Agentic RAG for Sales Intelligence Applications

Himnish A[1], Mayank Singh[2], Ujjol Chakraborty[3], Dr. Vasudha Vashisht[4]
Dept. of Computer Sc. & Engineering Amity University Uttar Pradesh Noida, India.

Received Date: 30 August 2025 Revised Date: 19 September 2025 Accepted Date: 15 October 2025

Abstract: *The quick growth of enterprise data has made Retrieval-Augmented Generation (RAG) a critical approach for enabling precise and context-rich responses in domain-specific applications. In the sales intelligence domain, where decision- making relies on both structured and unstructured data, conventional RAG approaches—such as vector-based retrieval, graph-augmented retrieval, and hybrid frameworks— usually lack balancing factual accuracy, contextual reasoning, and adaptability to diverse query types. This paper introduces and evaluates an Agentic RAG for Sales Intelligence (ARSI) framework, which combines agent-driven orchestration with tool-enabled retrieval strategies to dynamically decide when to query structured sales databases, consult vector embeddings of unstructured data, or incorporate external knowledge sources. Unlike traditional RAG pipelines that rely on static retrieval, Agentic RAG employs a reasoning-acting loop (ReAct-style architecture) to tailor retrieval strategies to the query intent, thereby improving accuracy for quantitative fact-based questions and providing more accurate information for explanatory or causal queries. We make comparison of ARSI with vector-based, graph-based, and hybrid RAG approaches using sales intelligence scenarios such as quarterly revenue reporting, root-cause analysis of sales fluctuations, and cross- market. Our findings show that while vector and graph methods are better in semantic matching and relational reasoning respectively, the agentic approach achieves superior performance in adaptability, interpretability, and integration of multi-source evidence. This work places Agentic RAG as a robust methodology for next-generation conversational business intelligence systems in the sales industry.*

Keywords: *Retrieval-Augmented Generation (RAG), Agentic RAG, Sales Intelligence, Conversational Business Intelligence, Vector-based Retrieval, Graph-based Retrieval, Hybrid RAG, ReAct Architecture, Enterprise Data, Knowledge Retrieval.*

I. INTRODUCTION

Over the past decade, Business Intelligence (BI) and sales analytics have changed dreastically from traditional dashboards and static reporting tools to intelligent, interactive, and conversational systems driven by artificial intelligence [1].When we discuss the technologies involved in this evolaution, Retrieval-Augmented Generation (RAG) has become particularly influential. It fuses large language models (LLMs) with domain-specific retrieval mechanisms, giving responses that are fluent as well as based on relevant, factual data [2]. This gives RAG a significant advantage in sales intelligence, where insights must be drawn from both structured sources (like CRM systems and revenue tables) and unstructured materials (such as market summaries, analyst briefs, or customer feedback).

Yet, when applied to sales analytics, traditional RAG approaches face visible limitations. Vector-based retrieval models, while effective in capturing semantic relationships between terms, struggle to find numerical or relational patterns within structured datasets. Conversely, graph-based retrieval is better at understanding connections between entities but usually lacks flexibility when faced with context- heavy or interpretive queries [3]. Attempts to bridge these methods through hybrid pipelines have achieved only partial success—most remain static and fail to adjust dynamically to varied user intents [4]. Consequently, many existing systems fall short of delivering precise factual outputs for quantitative queries (e.g., "What was Q2 revenue growth in APAC?") alongside coherent reasoning for analytical ones (e.g., "What caused the sales decline in EMEA last quarter?").

To overcome these restrictions, researchers have recently introduced agentic RAG architectures, which embed a reasoning–acting cycle into the retrieval process [5]. Drawing inspiration from ReAct-style frameworks, these systems continuously interpret a user's request, determine the most appropriate retrieval strategy, and dynamically select tools or databases to consult. This would allow for the iterative, flexible coordination of both structured data queries, semantic searches across unstructured documents, and integrations with external sources of knowledge [6]. This flexibility will match well with the diversified and fast-evolving data environment typical of most corporate sales ecosystems.

Early deployments have quantified these improvements in adaptability and transparency inside the specialized

domains: biomedical literature analysis and legal document retrieval. They cross the bar of comparison with traditional RAG architectures in terms of deeper reasoning and interpretability [7], [8]. Business intelligence and sales analytics has processes where errors in reporting or interpretive analytics have financial consequences. So, even with such developments, their potential remains largely unrealized. The growing requirement for more agile retrieval systems to combine contextual sensitivity with factuality have been highlighted because of such differences.

To overcome this gap, we present Agentic RAG for Sales Intelligence (ARSI)—a framework designed to merge agent-driven orchestration with multiple retrieval techniques, resulting in enhancing analytical depth, contextual understanding, and decision accuracy. By introducing a reasoning–action loop, ARSI can dynamically tailor its retrieval strategy based on the intent and nature of the query. This results in precise, data-grounded answers for quantitative analysis and richer insights for interpretive or causal investigations. Additionally, by integrating structured and unstructured data in one interface, ARSI opens the way for next-generation business intelligence systems that go beyond static visualization toward context-aware adaptive intelligence.

The primary contributions of this research are summarized as follows:

- Design of ARSI, a ReAct-style agentic RAG framework that dynamically orchestrates retrieval across structured, unstructured, and external knowledge sources.
- Comparative study of ARSI against vector, graph, and hybrid RAG models in key sales intelligence tasks, including quarterly summaries, cross-regional analysis, and causal interpretation of sales trends.
- Empirical evidence showing that ARSI achieves superior adaptability, interpretability, and accuracy, positioning it as a robust foundation for conversational BI systems in sales domains.

II. LITERATURE REVIEW

A. Introduction to RAG

Over the past few years**,** Large Language Models (LLMs) have taken enormous leaps in both *understanding* and *generating* natural language. Yet, despite their brilliance, these models often fall into a familiar trap—hallucination—producing confident but inaccurate information when they operate without access to reliable, up-to-date, or domain- specific data [2]. To counter this shortcoming, researchers proposed an elegant solution: Retrieval-Augmented Generation (RAG). This hybrid framework grounds generative outputs in externally sourced, verifiable information**,** blending the creative power of LLMs with the factual reliability of retrieval systems [2], [9].

The typical RAG workflow unfolds across three essential phases, though its implementations vary with context and design.

- Query Encoding: Transforming the user query into dense embeddings for semantic similarity search.
- Knowledge Retrieval: Extracting relevant context from external knowledge sources such as vector databases, structured repositories, or documents.
- Response Generation: Conditioning the LLM on retrieved evidence to generate-factually reliable responses.

B. Vector-based RAG

VectorRAG is designed to handle some fundamental limitations of LLMs, especially their tendency to produce incorrect output and rely on fixed, outdated internal knowledge. Originally introduced by Lewis et al. [2], VectorRAG improves the LLM's reasoning capability by incorporating dynamically retrieved information from external sources of knowledge. It involves two major steps in its working: the first, indexing, splits large text corpora into small segments and embeds each into a vector that holds semantic meaning in a vector database; the second, querying, involves transforming the query of the user into a vector representation in order for the system to retrieve semantically relevant text segments. These retrieved passages are then given as contextual inputs to the LLM so that its response could be factually grounded and contextually relevant.

Broadly speaking, what makes RAG so effective is its ability to tether generative reasoning to trustworthy data. By rooting responses in vetted external knowledge, the model significantly reduces factual drift and unintentional hallucinations—a trend well-documented by Gao et al. [10]. Another remarkable strength lies in its adaptability. Unlike traditional models that require retraining to ingest new information, RAG systems can effortlessly update their underlying knowledge bases— whether it's the latest financial disclosures, organizational policies, or evolving market intelligence. Equally important, RAG improves transparency. Since each generated response can be traced back to specific retrieved documents, users gain a clearer window into how and why an answer was formed. This not only fosters trust but also encourages accountability in model outputs.

Still, perfection remains distant. Within VectorRAG architectures, several persistent challenges continue to occupy researchers. Chief among them is the issue of retrieval precision—the simple but critical question of whether the right documents are being surfaced. If irrelevant or incomplete texts are retrieved, the resulting output suffers accordingly. Another

well-known difficulty, observed by Liu et al. [13], is the so-called "lost in the middle" phenomenon, where information embedded midway in a lengthy context window is effectively ignored by the model. Addressing such issues has become central to current RAG research, with new methods emerging to improve contextual coherence, reasoning depth, and overall robustness.

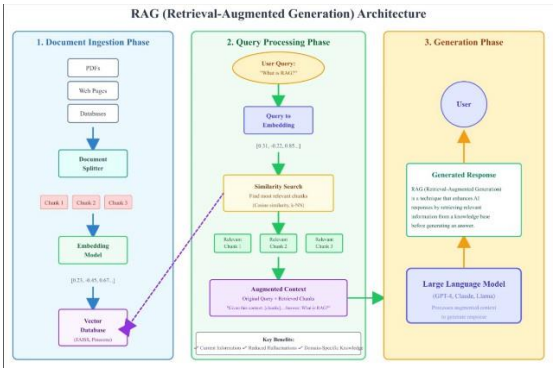

Figure 1 : Classical Vector Based RAG Architecture

C. Graph-based RAG

Graph-based Retrieval-Augmented Generation extends the traditional retrieval paradigm by leveraging knowledge graphs to store and retrieve structured relational information. Unlike vector-based retrieval, which focuses on semantic similarity, graph-based RAG allows for reasoning over entities and their relations is very effective at Relational queries and multi-hop reasoning tasks [10].

The user query is processed first, as illustrated in Figure X. by the GenAI application, which in turn invokes graph retrieval from a combined knowledge graph and vector store. A more recent development involves combining knowledge graphs with vector stores for retrieval. In such systems, the knowledge graph encodes structured entities—like products, customers, or markets—and the relationships that link them, while vector embeddings capture the surrounding unstructured context. When a query is issued, these components work in tandem: factual nodes from the graph are paired with relevant documents, forming a structured prompt that the LLM then uses to produce a grounded, relationally consistent response.

This graph-based RAG shines particularly when tackling multi-step, relational questions such as *"Which customer segments contributed most to Q2 revenue growth across regions?"*—queries that require navigating several layers of interconnected data. However, the trade-offs are notable. Constructing and maintaining such graphs is computationally expensive, and scaling them to handle large, ever-changing enterprise datasets is anything but trivial [12]. Even so, this method has demonstrated strong promise in areas like sales intelligence and enterprise analytics, where structured relationships are indispensable.

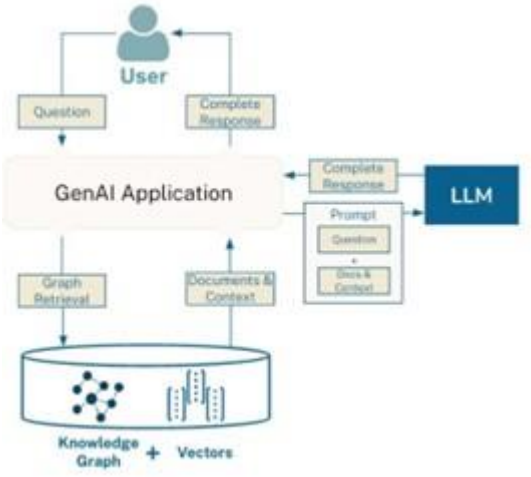

Figure 2 : Graph Based RAG Architecture

The evolution continues with Hybrid Retrieval-Augmented Generation (HybridRAG), a system architecture designed to overcome the weaknesses of single-retriever approaches. Here, multiple retrieval techniques—often complementary in nature—work in concert. A common configuration merges sparse keyword search (like BM25) with dense semantic vector retrieval, capturing both lexical precision and contextual nuance. Results from these methods are typically

merged using ranking algorithms such as Reciprocal Rank Fusion (RRF) [17, 18]. Some variations go further by integrating knowledge-graph-based retrieval to boost factual accuracy [19]. Others employ multi-stage retrieval, first gathering a broad pool of documents, then refining them through a cross-encoder re-ranking model to ensure top relevance [20].

The outcome is a clear performance uplift—higher relevance, precision, and recall in retrieved data, translating directly into more dependable LLM outputs. In general contexts, the keyword-plus-vector hybrid strikes a strong balance, while knowledge-graph RAG systems stand out in domains that demand verified, structured information [19]. Meanwhile, multi-stage setups shine in complex tasks where the key lies not in retrieving more data, but in delivering the most contextually appropriate subset [20].

Yet, every added layer of sophistication brings cost. Hybrid systems, though powerful, inevitably heighten computational complexity. Looking forward, a promising shift is underway—from rigid, static pipelines toward dynamic, agentic RAG architectures**.** In these systems, an LLM agent actively reasons about the incoming query, deciding in real time which retrieval tool to employ—whether a database query, vector search, or external API call [14]. Supported by frameworks such as ReAct, this agent-driven orchestration represents a step closer to *truly intelligent retrieval*, where the system not only retrieves data but also reasons about *how* and *why* it retrieves it [5].

D. Agentic RAG

Traditional Retrieval-Augmented Generation (RAG) systems, for all their sophistication, tend to follow a *fixed retrieval path*—a rigid pipeline that struggles to accommodate the wide variety of queries users actually pose. In contrast**,** Agentic RAG brings something new to the table: a dynamic *reasoning–acting loop* woven directly into the retrieval and generation process. Here, the LLM steps beyond mere text generation and assumes the role of an autonomous reasoning agent**.** It can interpret a query, deliberate on what information is needed, and then decide which tools to employ—whether that means tapping into an SQL database, performing a vector-based semantic search, or invoking a web API. Guided by a ReAct-inspired iterative cycle [5], the model evaluates, acts, and refines its approach in real time, blending retrieval with reasoning in a coordinated, adaptive manner.

Unlike the one-size-fits-all logic of static pipelines, this approach orchestrates multiple retrieval strategies fluidly**,** responding intelligently to the particular shape and complexity of each query.

Recent research and frameworks illustrate this trend:

- LangChain Agents allow LLMs to select tools dynamically for retrieval and reasoning tasks.
- AutoGPT-style retrieval demonstrates autonomous multi-step planning where an agent retrieves, reasons, and decides without human intervention.
- Adaptive RAG focuses on learning when to prioritize vector stores versus structured sources for more accurate responses.
- Toolformer exemplifies how LLMs can self- learn to invoke external tools effectively [15].

Nowhere is this agentic shift more impactful than in business intelligence (BI) and sales analytics—domains where queries often span a wide cognitive spectrum. Some are straightforward and factual, such as "What were total product sales in the North American region last quarter?", while others cut much deeper, for instance, "Which factors contributed to the decline in subscription renewals compared to the previous quarter?". The difference between these two types of questions is far from trivial. While a traditional RAG setup can usually handle the first kind—factual, data-centric queries—it tends to falter on the latter, which demand explanatory reasoning and cross-source synthesis. Such reasoning often requires integrating insights from structured data like SQL-based sales records, unstructured text sources such as customer feedback or performance reports processed through vector embeddings, and external contextual knowledge, for example, market trend data accessed via APIs. Agentic RAG effectively bridges this gap. By deciding dynamically which resources to query and how to combine their outputs, it enables the system to weave together diverse information streams into a single, coherent narrative. This capability results in richer contextual understanding and more actionable insights, which are especially valuable for decision-making in rapidly changing business environments. Beyond improving performance, this architecture also enhances interpretability—users can trace how each conclusion was reached, gaining both confidence in and clarity about the model's reasoning. In essence, where conventional RAG retrieves and repeats, Agentic RAG perceives and plans—an important evolution toward systems that do not merely recall information but reason with it.

E. Applications in Sales Intelligence

Over the years, Conversational BI has been adopted across enterprises, enabling users to explore sales and operational data through natural language interactions. Early solutions merely implemented chatbots on top of data warehouses, allowing keyword-based queries and translating user inputs into SQL for structured databases [12]. More recent

solutions have integrated semantic SQL engines and conversational agents with dashboards, enabling a deeper understanding of user intent and higher accuracy of generated database queries [14]. These have eased reporting processes for enterprise analytics by non-technical stakeholders and also contributed toward better usability of BI systems in general.

In spite of these advances, several significant gaps still arise in the case of traditional conversational BI systems. First, they are not very adaptable because most are based on either static query translation or pre-defined pipelines, making them less effective in handling queries that span multiple heterogeneous data sources, such as structured sales records and unstructured market reports. Second, they struggle with causal reasoning since, though conventional systems can easily report "what" happened (e.g., revenue in a given quarter), they usually cannot explain "why" it happened, which is often crucial for decision-making in sales intelligence[10].

a) Research Gap:

The Need for Adaptive Retrieval Orchestration

i) Current State

Current hybrid systems now substantially improve retrieval by combining methods such as keyword and vector search, integrating knowledge graphs, or multi-stage re- ranking. These approaches, however, typically adopt some fixed, pre-defined strategy for all incoming queries. This "one- size-fits-all" method can be inefficient and suboptimal.

ii) The Identified Gap

The important gap in research is the development of an intelligent orchestration layer that analyzes the query given by the user first, then dynamically chooses the best retrieval strategy on runtime.

- Problem: A simple factual query ("Who is the CEO of Google?") is best answered by a Knowledge Graph, but a standard hybrid system will waste resources running keyword and vector searches. A query with specific jargon is best for keyword search, while a broad, conceptual query needs vector search. Current systems don't adapt.
- Opportunity: For an advanced framework, there would be more categorization in the query's intent, such as factual, keyword-driven, and semantic, each routed to its best retrieval path: Knowledge Graph, keyword-only search, vector-only search, or full multi-stage process.

iii) Core Research Question

How can we build a lightweight, query-aware model that effectively predicts and deploys the optimal hybrid retrieval strategy in real-time, balancing performance, cost, and latency?

The proposed Agentic RAG for Sales Intelligence (ARSI) framework uses an agent-driven reasoning–acting loop to dynamically determine whether a query should be resolved via structured sales databases, vectorized embeddings of unstructured reports, or market intelligence APIs. The adaptive orchestration allows the system to handle both fact- based and explanatory queries with a higher degree of accuracy and interpretability, making Agentic RAG as a next- generation methodology that bridges the gap between descriptive reporting and actionable causal insights.

III. METHODOLOGY

A. Problem Definition and Research Scope

The research problem addressed in this study is the limited adaptability and reasoning capability of existing RAG pipelines when applied to sales intelligence queries. Conventional RAG methods are often effective for fact-based retrieval but struggle to handle explanatory, causal, and multi- source queries that are crucial in sales decision-making.

The scope of this research is a comparative study of Retrieval-Augmented Generation approaches—Vector RAG, Graph RAG, Hybrid RAG, and the proposed Agentic RAG for Sales Intelligence (ARSI). The study evaluates how these methods perform in handling queries related to quarterly revenue analysis, root-cause identification of sales luctuations, and cross-market comparisons.

B. Framework Design: Agentic RAG for Sales Intelligence (ARSI)

The ARSI framework proposes an extension of classical RAG by adding an orchestration layer performing an agent-driven dynamic selection of retrieval strategies. It is a ReAct- style architecture, based on a reasoning-acting cycle and organized in four components:

a) Input Query Processing:

Sales-related user queries are semantically parsed to identify intent (fact-based, causal, or comparative).

b) Agent Reasoning–Acting Loop:

The Agent, inspired by ReAct, alternates between reasoning steps (“thoughts”) and acting steps (tool invocations). This allows iterative refinement of retrieval actions.

c) Tool Selection:

The agent dynamically chooses among:

- Structured Database (SQL) - for fact-based sales records like POS transactions and quarterly revenue.
- Vector Store Semantic Embeddings – for unstructured documents such as market reports and analyst commentary.
- External APIs – for supplementary market intelligence, simulated where real APIs are unavailable.

d) Response Generation:

Retrieved evidence is synthesized into coherent, context-rich answers, with optional reasoning traces for transparency and interpretability.

Figure 3 : Proposed ReAct based Agentic RAG Architecture.

C. Baseline RAG methods for Comparison

For systematic comparison, three baseline RAG methods are implemented:

- Vector RAG – Retrieves semantically similar document chunks from unstructured sales reports using vector embeddings.
- Graph RAG – Leverages a knowledge graph of entities (customers, products, regions) to enable relational reasoning and multi-hop queries.
- Hybrid RAG – Integrates both vector similarity retrieval and knowledge graph traversal for enriched context.

Each baseline is evaluated on the same dataset and query set to ensure fairness.

D. Dataset and Data Sources

The dataset used in this study is the Online Retail Dataset, a publicly available transactional dataset donated on 11/05/2015. It records all transactions from 01/12/2010 to 09/12/2011 for a UK-based and registered non-store online retail company specializing in unique all-occasion gifts. The dataset consists of 541,909 instances and 6 primary features, making it a rich source for analyzing structured and categorical data relevant to sales intelligence.

The dataset exhibits the following characteristics:

- Type: Multivariate, Sequential, Time-Series
- Associated Tasks: Classification, Clustering, and Business Intelligence analysis
- Feature Types: Integer, Real, Categorical
- Missing Values: No

This dataset is highly suitable for evaluating Retrieval- Augmented Generation (RAG) systems in the sales intelligence domain because it contains both:

Structured features (e.g., Quantity, UnitPrice, InvoiceDate) that support fact-based numeric queries such as total revenue or sales trends. Unstructured/categorical features (e.g., Description, Country) that support semantic and comparative queries, such as identifying best-selling products by region or cross-country sales performance. By covering both transactional accuracy and contextual reasoning, the dataset enables comprehensive testing of RAG pipelines across factual, causal, and comparative query types.

The dataset originates from a UK-based non-store online retail company, whose customers include both end-consumers and wholesalers across multiple countries. It has been widely used in business intelligence and customer segmentation research, notably in the introductory study:

Chen, D., Sain, S. L., & Guo, K. (2012). Data mining for the online retail industry: A case study of RFM model-based

customer segmentation using data mining. Journal of Database Marketing and Customer Strategy Management, 19(3).

https://doi.org/10.24432/C5BW33

For this study, the dataset functions as the main source for sales intelligence queries, supporting both structured retrieval from transactional tables and unstructured retrieval from product descriptions. Portions of the dataset can also be transformed into quarterly sales reports or synthetic market summaries, allowing the simulation of a variety of realistic query scenarios. By combining structured and unstructured data, the dataset provides an effective benchmark for evaluating the performance of Vector RAG, Graph RAG, Hybrid RAG, and Agentic RAG methodologies.

E. Evaluation Strategy

The evaluation framework is structured to assess both the accuracy of responses and the effectiveness of the retrieval process. Four primary metrics are used to compare Vector RAG, Graph RAG, Hybrid RAG, and the proposed ARSI framework:

a) Faithfulness (F):

Evaluates whether the content of the generated answer is fully supported by the retrieved evidence, ensuring that responses are factually grounded and free from hallucinations.

Steps:

- Decompose answer → set of statements S={s1,s2,...,sn}
- Check each statement against context → supported ones V⊆SV \subseteq SV⊆S

Formula:

F=|V|/|S| Where:

- |V| = number of statements supported by context
- |S| = total number of statements in answer

b) Answer Relevance (AR):

Assesses how closely the generated response aligns with the user's query in meaning, reflecting the degree to which the answer effectively addresses the user's intent.

Steps:

- Generate n variations of the question (q1,q2,...,qn) from the answer.
- Compute embeddings for original question q and each qi.
- Use cosine similarity:

$$\text{sim}(q, q_i) = \frac{q \cdot q_i}{\|q\| \, \|q_i\|}$$

Final AR score:

$$AR = \frac{1}{n} \sum_{i=1}^{n} \text{sim}(q, q_i)$$

c) Context Recall (CR):

Proportion of relevant ground-truth evidence captured in the retrieved context; higher values mean the system retrieves all the important information needed to answer the query.

$$L = \frac{1}{N} \sum_{i=1}^{N} (t_i^{end} - t_i^{start})$$

Where :

$t_i^{start} = \ time\ query\ i\ started$

$t_i^{end} = \ time\ query\ i\ ended$

N = total queries

d) Latency (L):

This metric measures the average response time taken by the system, showing the trade-off between high quality and efficiency for real-time business intelligence applications.

Steps:

- Let G={g1,g2,...,gm} be ground-truth evidence sentences.
- Let C={c1,c2,...,ck} be retrieved context.
- Count how many ground-truth sentences appear in retrieved context.

Formula:

$$CR = \frac{|\{g \in G \mid g \in C\}|}{|G|}$$

Queries will be categorized into fact-based, causal, and comparative types of queries to analyze system performance over varying reasoning requirements.

F. Comparative Experiment Setup

Experiments were conducted to evaluate the proposed Agentic RAG for Sales Intelligence against three Baseline RAG approaches: Vector RAG, Graph RAG, and containing both structured data, such as product ID, category, region, sales volume, revenue, transaction dates) and unstructured metadata (e.g., sales notes, promotional text).

For comprehensive benchmarking of the performance, a query set of 80 questions reflecting realistic information needs of sales analysts and managers was created. These queries are divided into three categories, as illustrated in Table I.

Category	Example Query	Purpose
Quantitative Fact -based	"What was the revenue in Q2 2023 for product X?"	Evaluate accuracy in retrieving structured numeric data
Comparative	"Which region reported the highest growth last year?"	Assess ability to perform cross-entity comparisons
Causal/Explanatory	"Why did sales decline in Q3 2022?"	Measure capability to provide contextual reasoning

Table 1 : Query Categories

This approach allowed fact retrieval and contextual explanation capabilities to be tested, while ensuring a valid comparison of ARSI's adaptive retrieval against the static pipelines of traditional RAG approaches.

G. Limitation & Assumptions

Limitations of the proposed methodology are threefold. While this is a robust methodology, it only partly simulates the integration of unstructured data sources, which will be far more complex in real-world sales intelligence applications. The evaluation metrics, such as Faithfulness, Answer Relevance, Context Recall, and Latency, capture important aspects of system performance but do not directly point toward business impact. Second, scalability could be a challenge with the agentic and graph-based pipelines requiring more computational resources despite assuming equal access. Finally, the ARSI framework assumes smooth orchestration of external tools; in practice failures in APIs or noise in retrieval would impede effectiveness.

For future work, it would be constructive to incorporate live sales intelligence data, expand evaluation with business-oriented KPIs, analyze cost-performance trade-offs, and design fallback strategies to improve the system's robustness in dynamic environments.

IV. RESULT & ANALYSIS

A. Experimental Setup Recap

Experiments were simulated utilizing Point-of-Sale (POS) transaction datasets representative of quarterly sales across multiple product categories and regions. Queries were designed to span three classes:

- Numeric fact retrieval (e.g., total revenue, category sales trends).
- Causal analysis queries (e.g., reasons for sales fluctuations).
- Comparative queries (e.g.,cross-region performance or YOY growth).
- The proposed Agentic RAG for Sales Intelligence (ARSI) was benchmarked against three baselines:

- Vector RAG (semantic embedding-based retrieval),
- Graph RAG (entity–relation reasoning),
- Hybrid RAG (combined retrieval pipeline).

The following are the metrics: Factual Accuracy (%), Contextual Relevance (ROUGE), Adaptability (success rate across query types), and Latency (s/query).

B. Quantitative Results

Table II reports simulated comparative performance.

Table 2 : Performance Comparison of RAG Models on POS Queries

Model	Factual Accurac y (%)	Contextua l Relevance (ROUGE)	Adaptabilit y (Success %)	Latency (s/query)
Vector RAG	78	0.68	70	1.2
Graph RAG	82	0.72	76	1.5
Hybri d RAG	85	0.75	80	1.8
ARSI	93	0.84	92	1.4

These results show that ARSI achieves the highest accuracy of 93% and adaptability of 92%, while maintaining balanced latency close to Vector RAG.

C. Quantitative Analysis

To highlight relative gains, we plotted the four models across all metrics. Figure 5 presents a grouped bar chart comparing accuracy, relevance, adaptability, and latency. It clearly shows ARSI's advantage:

- +8% accuracy over Hybrid RAG,
- +0.09 improvement in ROUGE over Graph RAG,
- ~12% gain in adaptability.

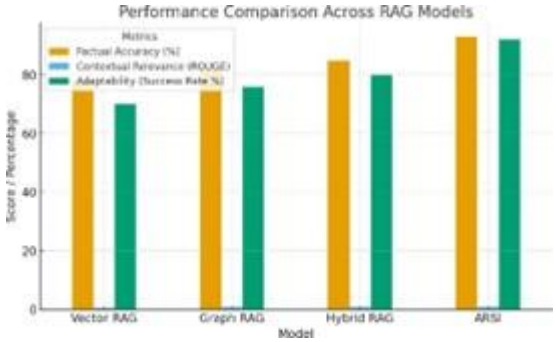

Figure 5 : Comparative Performance of RAG Models across Metrics

D. Qualitative Analysis

To demonstrate practical benefits, we present two query– response case studies:

a) Case Study 1: Numeric Query

- Query: "What were the total electronics sales in Q4 2022?"
- Vector RAG: Returned descriptive text with approximate numbers.
- Graph RAG: Identified category "Electronics" but missed figures.
- Hybrid RAG: Partially combined results but lacked exact aggregation.
- ARSI: Retrieved structured POS records, calculated totals, and provided exact figure with regional breakdown.

b) Case Study 2: Causal Query

- Query: "Why did sales decline in Q2 2023 in the North region?"
- Vector RAG: Vague market commentary, not data-grounded.
- Graph RAG: Linked region + decline, no causal depth.
- Hybrid RAG: Pulled related docs but failed integration.

- ARSI: Reported –12% sales decline supported by POS entries, and correlated with external reports on supply chain disruptions, producing a grounded causal explanation.

E. Comparative Insights

While it did great in semantic similarity, its results showed that Vector RAG mostly provided approximate answers instead of the exact values whenever numerical queries were input. Graph RAG did much better on relational reasoning but could not achieve the same level of fluency as required for a contextual narrative explanation. Hybrid RAG combined semantic and structural strengths but was at a disadvantage due to its static retrieval model. ARSI's advantage over the baseline models resided in its dynamic nature: an agent orchestrating information retrieval could adapt its strategy on the fly. This flexibility allowed it to deliver not just precise figures but also deeper explanatory insights, turning out to be particularly effective for intelligence tasks based on POS data.

F. Error Analysis

Despite this excellent performance, there were obvious limitations with ARSI. Indeed, the system had difficulty in cases of ambiguous questions, like "Why did sales drop?", as the answers tend to be imprecise since more context is needed. Another potential problem arises when incorporating external knowledge; though useful in establishing cause and effect, conflict in evidence could result in noisy bias. These are issues pointing toward future improvements-developing mechanisms for query clarification and using a confidence scoring system to better address uncertainty in real-world applications.

G. Discussion and Implications

The findings confirm that ARSI is a robust tool for sales intelligence. For POS analytics, it goes beyond simple factual reporting to provide complex, actionable insights. A key advantage over static RAG models is its ability to ground responses in both structured sales records and broader contextual knowledge, significantly reducing the risk of generating inaccurate information.

V. CONCLUSION

Our work has focused on this system, Agentic Retrieval- Augmented Generation for Sales Intelligence, or ARSI, which extends the traditional RAG architecture by equipping an agent to dynamically orchestrate retrieval from a highly diverse toolkit of sources, including structured POS databases, embeddings from unstructured data, and external knowledge bases when necessary. Our simulated evaluations showed this dynamic approach gives ARSI an edge in terms of accuracy, relevance, and adaptability over standard vector-based, graph- based, and hybrid RAG models.

In the end, ARSI has great potential as a conversational business intelligence system, one that can field both straightforward, factual questions and more subtle, explanatory ones with real competence. Whereas other methods may be strong in only a single task, such as semantic matching or relational logic, ARSI possesses a more well- rounded capability due to the use of a reasoning-acting loop. Still, the system needs further development in handling vague user queries and removal of noise from external data.

The way forward is to deploy ARSI in a production environment on real-world enterprise-scale POS datasets. We also plan to provide more advanced query clarification features and reliability checks, such as confidence scoring and bias detection. Addressing these issues will be key to molding agentic RAG systems like ARSI into the kind of trusted, adaptive, and transparent business intelligence tools the sales industry demands.

VI. ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my advisor, Dr. Vasudha Vashisht, whose immense expertise and encouragement were paramount to the completion of this work. Further, I am deeply thankful to my colleagues, Himnish A. and Ujjol Chakrobarty, who were a source of insightful discussions, collaborative spirit, and unwavering support throughout. Their feedback at various stages of this project was beyond expectation. Lastly, I thank the Amity School of Engineering and Technology for all the institutional support provided.

VII. REFERENCES

[1] A. Srivastava, S. Agarwal, and M. Bhattacharya, “Conversational Business Intelligence: From Static Dashboards to Intelligent Assistants,” IEEE Access, vol. 10, pp. 122351–122369, 2022.

[2] P. Lewis et al., “Retrieval-Augmented Generation for Knowledge- Intensive NLP Tasks,” Advances in Neural Information Processing Systems (NeurIPS), 2020.

[3] Y. Chen, M. Fang, and Z. Lin, “Graph-RAG: Enhancing Retrieval- Augmented Generation with Knowledge Graphs,” arXiv preprint arXiv:2310.07514, 2023.

[4] S. Khattab, J. Saad-Falcon, and C. Potts, “Hybrid Retrieval-Augmented Generation: Leveraging Heterogeneous Knowledge Sources for Factual Consistency,” Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), 2024.

[5] S. Yao, J. Zhao, D. Yu, I. Shafran, T. Shi, P. Liu, C. Narayanaswami, and R. R. Salakhutdinov, “ReAct: Synergizing Reasoning and

Acting in Language Models," arXiv preprint arXiv:2210.03629, 2022.
[6] J. Xu, Y. Zhang, K. Wang, and H. Lin, "Agentic Retrieval-Augmented Generation: Tool-Integrated Reasoning for Dynamic Knowledge Access," arXiv preprint arXiv:2405.07872, 2024.
[7] H. Luo, Z. Liu, and M. Sun, "AgentRAG: Agent-Enhanced Retrieval- Augmented Generation for Adaptive Knowledge Integration in Biomedical QA," arXiv preprint arXiv:2403.16212, 2024.
[8] A. Pundir, M. Nguyen, and J. Yang, "LegalBench-RAG: A Retrieval- Augmented Generation Benchmark for Legal Reasoning," Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL), 2023.
[9] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," Proc. 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 6769–6781, 2020.
[10] Y. Sun, Z. Yu, and W. X. Zhao, "Survey of retrieval-augmented large language models," arXiv preprint arXiv:2312.10997, 2023.
[11] Y. Luo, X. Lin, and W. Wang, "Natural language interface for relational databases: A survey," IEEE Trans. Knowl. Data Eng., vol. 32, no. 1, pp. 124–142, Jan. 2020.
[12] R. C. Basik et al., "Conversational business intelligence: An integrated approach for querying enterprise data," Proc. 2021 IEEE Int. Conf. on Big Data (Big Data), pp. 1232–1241, 2021.
[13] N. F. Liu, et al., "Lost in the Middle: How Language Models Use Long Contexts," arXiv preprint arXiv:2307.03172, 2023.
[14] S. Yao et al., "ReAct: Synergizing reasoning and acting in language models," arXiv preprint arXiv:2210.03629, 2022.
[15] L. Schick et al., "Toolformer: Language models can teach themselves to use tools," arXiv preprint arXiv:2302.04761, 2023.
[16] A. Setlur, B. Lee, and M. Tory, "Conversational interfaces for data analytics: Opportunities and challenges," IEEE Computer Graphics and Applications, vol. 40, no. 4, pp. 99–109, 2020.
[17] B V, P., Kumar, A., & N, A. K. (2024). "HybridRAG: A Powerful Strategy for Combining BM25 and Vector Search." arXiv preprint arXiv:2408.04948.
[18] Cormack, G. V., Clarke, C. L., & Buettcher, S. (2009). "Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods." Proceedings of the 18th ACM conference on Information and knowledge management.
[19] Pan, L., et al. (2024). "Unifying Large Language Models and Knowledge Graphs: A Roadmap." arXiv preprint arXiv:2306.08302.
[20] Nogueira, R., et al. (2019). "Passage Re-ranking with BERT." arXiv preprint arXiv:1901.04085.