NextGen Scientific Publication
American International Journal of Computer Science and Technology

Volume 6 Issue 3, Pg. No. 114-126, AIJCST-V6I3P109, 2024
https://doi.org/10.63282/3117-5481/AIJCST-V6I3P109

Original Article

Retrieval-Augmented Generation for Trusted Customer Intelligence: A Scalable Enterprise Architecture for CRM Knowledge Management

***Achuta Krishna Kishore Varma Alluri**

Salesforce CRM Lead, Informa Support Services Inc, Des Plaines, Illinois, United States.

Abstract:

Enterprise customer relationship management systems increasingly operate as repositories of fragmented customer knowledge rather than as unified intelligence platforms. Sales representatives, service agents, customer-success teams, marketing analysts, and executives rely on CRM data, call transcripts, support tickets, contracts, product documentation, knowledge articles, renewal histories, and external market signals, yet these assets are often distributed across incompatible schemas, permissions, and narrative formats. Large language models offer new opportunities for conversational customer intelligence, but their direct use in CRM environments introduces unacceptable risks, including hallucinated recommendations, privacy exposure, stale knowledge, weak auditability, and limited alignment with enterprise decision governance. This paper proposes a scalable retrieval-augmented generation architecture for trusted customer intelligence in enterprise CRM knowledge management. The proposed framework integrates hybrid retrieval, semantic indexing, policy-aware document ingestion, customer-entity resolution, vector and graph-based knowledge representations, evidence-grounded generation, human-in-the-loop feedback, and trust-oriented observability. The architecture is designed to support account planning, opportunity qualification, customer service resolution, churn-risk explanation, sales enablement, executive decision support, and knowledge reuse across customer operations. A design-science methodology is adopted to formalize the artifact, decompose architectural layers, define trust controls, and establish evaluation metrics for retrieval quality, answer faithfulness, decision usefulness, latency, scalability, compliance, and user adoption. The analytical discussion shows that enterprise RAG is not merely an LLM enhancement pattern but a socio-technical knowledge-governance architecture that transforms CRM from a passive transaction system into a source-grounded intelligence environment. The study contributes a conceptual model, evaluation framework, and implementation roadmap for organizations seeking to deploy generative AI in customer-facing functions without compromising factual reliability, security, or managerial accountability.

Keywords:

Retrieval-Augmented Generation, Customer Relationship Management, CRM Knowledge Management, Customer Intelligence, Enterprise AI Architecture, Trusted AI, Sales Intelligence, Knowledge Graphs, Large Language Models, Human-AI Collaboration.

Article History:

***Received:** 28.03.2024*

***Revised:** 02.05.2024*

***Accepted:** 13.05.2024*

***Published:** 21.05.2024*

1. Introduction

Customer relationship management has historically been framed as a strategic, cross-functional approach for creating customer value through coordinated processes, information flows, and organizational capabilities rather than as a narrow software deployment.

In enterprise practice, however, CRM platforms frequently become fragmented operational systems in which structured fields, account notes, product histories, service interactions, email records, renewal data, and customer feedback remain only partially connected. This fragmentation weakens the ability of sales and customer operations teams to answer complex questions such as why an opportunity is stalled, which support incidents threaten renewal, how a customer's prior objections relate to current product adoption, or what evidence justifies a next-best action. A strategic CRM architecture must therefore connect customer knowledge, process execution, and performance assessment rather than simply store customer records [1].

The emergence of large language models has intensified expectations that CRM systems can become conversational intelligence platforms. Account executives may ask for an account briefing before a meeting, service leaders may request a root-cause summary of recurring complaints, and executives may ask for revenue-risk narratives across segments. Business intelligence research has long shown that organizational value arises when data, analytics, and decision processes are integrated into managerial action rather than treated as isolated reporting infrastructure [11]. Generative AI extends this principle by allowing users to interact with enterprise knowledge through natural language, but it also changes the risk profile of CRM analytics because fluent generated text may be persuasive even when unsupported by verifiable evidence.

The central challenge is that customer intelligence is both knowledge-intensive and trust-sensitive. A sales recommendation may influence pricing, renewal strategy, customer escalation, or contractual negotiation. A service response may expose regulated information or misrepresent an entitlement. A customer-success summary may incorrectly attribute churn risk to product dissatisfaction when the underlying evidence points to onboarding delay, billing friction, or unresolved integration issues. Knowledge-management scholarship emphasizes that organizational knowledge is not simply data; it is contextualized, experience-based, and action-oriented, requiring mechanisms for capture, validation, transfer, and application [23]. CRM intelligence architectures must therefore preserve provenance, context, access control, and interpretability as first-order design requirements.

Retrieval-augmented generation provides a promising foundation because it combines generative models with non-parametric enterprise knowledge retrieved at inference time. Instead of relying only on model parameters, a RAG system retrieves relevant documents, passages, database records, or knowledge-graph facts, then conditions generation on that evidence. This pattern is especially suitable for CRM environments where knowledge changes continuously through new calls, support tickets, emails, product releases, contracts, and customer interactions. Yet a naive RAG implementation is insufficient for enterprise customer intelligence. Without robust ingestion, retrieval governance, evidence ranking, permission filtering, answer verification, and feedback loops, RAG may simply produce more polished versions of existing CRM inconsistency.

This paper proposes a scalable enterprise architecture for retrieval-augmented CRM knowledge management. The architecture is designed around six principles: evidence grounding, customer-context preservation, policy-aware retrieval, human-AI collaboration, continuous evaluation, and operational scalability. It treats the CRM copilot or conversational assistant not as a detached chatbot but as a governed decision-support layer embedded in sales, service, marketing, and customer-success workflows. The framework draws from decision-intelligence thinking, where AI-enabled outputs must be connected to governance, lifecycle management, and architecture-centered project control rather than treated as isolated predictions [2].

The contribution of this paper is threefold. First, it formulates a trusted customer-intelligence problem space in which CRM knowledge management, generative AI, retrieval systems, and organizational governance intersect. Second, it proposes a layered enterprise RAG architecture that integrates data ingestion, semantic enrichment, vector indexing, graph-aware retrieval, policy enforcement, prompt orchestration, evidence-grounded generation, and trust monitoring. Third, it defines evaluation criteria for measuring retrieval precision, answer faithfulness, decision usefulness, compliance integrity, latency, adoption, and business impact. The paper is intentionally conceptual and analytical: it does not claim production deployment results from a single organization, but it provides a rigorous design framework suitable for empirical validation, prototyping, and enterprise implementation.

Finally, the proposed architecture responds to a broader transformation in enterprise AI. Foundation models and multimodal systems demonstrate powerful language and reasoning abilities, but enterprise deployment depends on whether those capabilities can be constrained, audited, updated, and aligned with domain-specific knowledge. The GPT-4 technical report illustrates the breadth of modern generative model capabilities, yet CRM environments require grounded generation that remains accountable to enterprise facts,

access policies, and customer-specific evidence rather than generalized model fluency [25]. Trusted customer intelligence therefore depends less on unrestricted model scale than on architectural integration between language models and governed knowledge systems.

2. Background and Related Work

Knowledge management provides the conceptual foundation for enterprise RAG in CRM because customer intelligence depends on capturing, organizing, sharing, and applying knowledge across organizational boundaries. Alavi and Leidner define knowledge-management systems as information systems that support the creation, storage, retrieval, transfer, and application of organizational knowledge, emphasizing that knowledge can be understood as object, process, capability, access to information, or justified belief [4]. This perspective is highly relevant to CRM because customer knowledge is not limited to static attributes; it includes tacit sales insights, service histories, contractual interpretations, product usage patterns, and contextual narratives about customer intent.

Traditional CRM analytics has relied heavily on structured data mining, segmentation, classification, association analysis, and predictive modeling. Ngai, Xiu, and Chau's literature review demonstrates the long-standing role of data mining in CRM, particularly for retention, identification, attraction, and development of customers [6]. However, contemporary CRM knowledge now includes large volumes of unstructured and semi-structured information such as call transcripts, meeting notes, emails, knowledge-base articles, customer complaints, and implementation documents. These artifacts contain valuable signals but are difficult to operationalize through conventional dashboards or field-based CRM analytics alone.

CRM implementation research also shows that technology adoption depends on alignment among people, processes, and systems. Zablah, Bellenger, and Johnston argue that CRM acceptance is shaped by the fit between employees, processes, and CRM technology rather than by technical functionality alone [18]. This insight is crucial for RAG-based CRM architectures because even highly accurate generated outputs may fail if users do not understand the evidence, cannot challenge the recommendation, or perceive the assistant as disrupting established sales and service workflows. Human-AI collaboration must therefore be designed into the architecture through feedback mechanisms, role-specific interfaces, and explainable evidence presentation.

The technical foundation of modern RAG is rooted in transformer-based language modeling. The Transformer architecture replaced recurrent and convolutional sequence modeling with self-attention, enabling scalable representation learning across long token sequences and forming the basis for contemporary language models [8]. Transformer-based models made it possible to build systems capable of summarization, question answering, semantic search, and dialogue, all of which are central to customer intelligence. In CRM settings, these capabilities can convert scattered enterprise knowledge into conversational interfaces, but the transformer mechanism alone does not guarantee factual grounding or domain correctness.

Bidirectional pretraining further advanced language understanding by enabling models to condition on both left and right context during representation learning. BERT demonstrated that pretrained transformer representations could be fine-tuned for question answering, language inference, and other natural language understanding tasks with strong performance [5]. For CRM knowledge management, BERT-style encoders and later embedding models are important because they support semantic retrieval over customer notes, knowledge articles, contracts, and service documentation. Semantic retrieval can find conceptually relevant passages even when users and documents use different wording, such as "renewal blocker," "procurement delay," and "contracting risk."

Large autoregressive language models expanded the role of generative systems by showing strong few-shot and task-agnostic performance. Brown et al. demonstrated that sufficiently large language models could perform many tasks from prompts without task-specific fine-tuning [17]. In CRM applications, this means an assistant can summarize a support escalation, draft an account plan, or transform a call transcript into action items. Nevertheless, few-shot language ability is not equivalent to enterprise reliability. Customer intelligence requires that generated text be traceable to current customer records and authorized knowledge sources rather than inferred from broad pretraining.

Retrieval-augmented generation directly addresses this limitation by combining parametric memory with retrieved non-parametric memory. Lewis et al. introduced RAG for knowledge-intensive NLP tasks, showing that retrieval can improve generation by conditioning outputs on external documents [3]. The enterprise implication is significant: CRM copilots can access current account records, product documentation, entitlement policies, and support histories at inference time, reducing reliance on static model memory.

This makes RAG particularly suitable for environments in which customer information changes daily and where stale or hallucinated knowledge can have material business consequences.

Dense retrieval is a key component of RAG because it enables semantic matching between user queries and relevant passages. Karpukhin et al. showed that dense passage retrieval could outperform traditional sparse retrieval for open-domain question answering by mapping questions and passages into a learned embedding space [9]. In CRM, dense retrieval can identify relevant support issues even when terminology varies across teams, customers, and product lines. However, enterprise CRM retrieval should not rely exclusively on dense vectors; it must often combine keyword search, metadata filters, graph relationships, access-control rules, recency weighting, and domain-specific reranking.

Recent RAG surveys distinguish naive, advanced, and modular RAG patterns, emphasizing retrieval, augmentation, generation, query transformation, indexing, reranking, and evaluation as separate design concerns [16]. This modular perspective is important for enterprise architecture because CRM use cases vary widely. A service-agent assistant may require low-latency policy retrieval; a sales-intelligence assistant may need account-level synthesis across transcripts and opportunities; an executive assistant may need aggregated trend analysis with strict source attribution. A modular RAG architecture allows enterprises to configure retrieval, reasoning, and generation pipelines according to task risk and operational constraints.

A major motivation for RAG is hallucination mitigation. Hallucination in natural language generation refers to output that appears fluent but is unsupported, unfaithful, or factually incorrect. Ji et al. provide a comprehensive survey of hallucination causes, measurement, and mitigation across summarization, dialogue, question answering, and other generation tasks [12]. In CRM, hallucination is not merely a linguistic defect; it can become a business-risk event. A hallucinated contract clause, unsupported customer complaint, or invented product capability may damage customer trust, create compliance exposure, or distort managerial decisions.

Retrieval has been shown to reduce hallucination in knowledge-grounded conversation by providing external evidence during generation. Shuster et al. demonstrate that neural retrieval-in-the-loop architectures can reduce hallucinations in conversational systems [15]. This finding supports the use of RAG in customer service and sales contexts, where conversational coherence must be combined with factual grounding. However, retrieval can only reduce hallucination if the retrieved evidence is relevant, current, authorized, and properly used by the generator. Poor retrieval can create false confidence by supplying plausible but irrelevant passages.

Another challenge is determining when retrieval is needed. Mallen et al. show that language models struggle with long-tail factual knowledge and that retrieval augmentation is especially useful when parametric memory is unreliable [20]. CRM knowledge is overwhelmingly long-tail: each customer has unique contracts, histories, relationships, product configurations, complaints, and negotiation dynamics. Therefore, enterprise CRM copilots should default toward retrieval for customer-specific tasks, but they may use adaptive retrieval to reduce latency and cost for generic drafting or formatting tasks that do not require proprietary customer facts.

The architecture must also account for long-context limitations. Liu et al. show that language models may fail to robustly use information placed in the middle of long contexts, even when context windows are large [24]. This result matters for CRM because a naive strategy of stuffing all account notes, tickets, transcripts, and emails into a long prompt can degrade reliability. Enterprise RAG should instead retrieve focused evidence, structure context windows, prioritize critical passages, and use hierarchical summarization when customer histories exceed prompt budgets.

Trustworthy AI frameworks provide governance requirements for enterprise deployment. The NIST AI Risk Management Framework identifies characteristics such as validity, reliability, safety, security, resilience, accountability, transparency, explainability, privacy enhancement, and fairness [10]. These characteristics map directly to CRM RAG design. A trusted customer-intelligence assistant must be reliable under changing data, secure under role-based access, transparent about evidence, privacy-preserving for customer data, and accountable through audit logs. Trust cannot be added after model deployment; it must be embedded into architecture, evaluation, and operating procedures.

European trustworthy AI guidance similarly emphasizes human agency, technical robustness, privacy and data governance, transparency, fairness, societal well-being, and accountability [21]. In CRM environments, human agency means that AI recommendations should support, not replace, sales and service judgment. Transparency means that generated outputs should include

evidence and uncertainty. Privacy and governance mean that sensitive customer records should be retrieved only under authorized contexts. These requirements reinforce the need for a socio-technical architecture rather than a purely model-centric implementation.

Prior work on AI-enabled software and decision systems also supports the need for lifecycle governance. Gunda's discussion of machine learning's expanding role in software development highlights the movement from isolated automation toward intelligent systems embedded across software lifecycle and organizational workflows [7]. This perspective is relevant to CRM RAG because enterprise copilots are not one-time analytics models. They require continuous monitoring, prompt versioning, retrieval evaluation, governance review, and adaptation as CRM schemas, customer policies, and organizational practices evolve.

3. Proposed Framework / Methodology

This paper adopts a design-science research orientation because the research objective is to develop and justify an artifact: a scalable enterprise RAG architecture for trusted customer intelligence. Design science is appropriate when knowledge is generated through the construction and evaluation of innovative information-systems artifacts that address relevant organizational problems [14]. The artifact proposed here is not a single software component but a reference architecture composed of data, retrieval, generation, governance, evaluation, and human-feedback layers. The methodology therefore combines problem identification, requirement derivation, architectural design, evaluation criteria, and analytical validation.

The first methodological step is to define the problem environment. In enterprise CRM, customer knowledge is distributed across transactional CRM tables, customer-service platforms, call-center transcripts, email systems, knowledge bases, product documentation, contracts, renewal notes, marketing systems, and analytics warehouses. These sources differ in freshness, structure, ownership, sensitivity, and reliability. A trusted RAG architecture must treat each source as a governed knowledge asset with metadata, access policy, lineage, and quality indicators. The ingestion pipeline must capture not only document text but also customer identifiers, account hierarchy, opportunity references, document timestamps, author roles, data classifications, and applicable retention policies.

The second step is to formalize user tasks and decision contexts. Customer intelligence tasks differ in risk and evidence requirements. Low-risk tasks include drafting meeting summaries, rewriting notes, or creating follow-up emails from authorized context. Medium-risk tasks include summarizing account health, identifying customer objections, and comparing feature requests. High-risk tasks include renewal-risk recommendations, pricing justification, contractual interpretation, compliance-sensitive service responses, and executive escalation narratives. The proposed methodology assigns each task to a risk tier that determines retrieval strictness, required citations, approval workflow, logging depth, and whether human verification is mandatory before output is used externally.

The third step is knowledge normalization. Enterprise CRM data contains duplicates, inconsistent naming, incomplete notes, obsolete documents, and conflicting interpretations. Therefore, the framework includes customer-entity resolution to connect contacts, accounts, opportunities, tickets, contracts, and communication records. It also includes document canonicalization, where unstructured assets are converted into standardized units with metadata fields such as source system, author, business process, customer entity, effective date, confidentiality level, and quality score. This step is essential because retrieval performance depends on the quality of indexed knowledge, not only on the embedding model or vector database.

The fourth step is policy-aware chunking and representation. Unlike generic document search, CRM retrieval requires chunks that preserve business meaning. A contract clause, ticket resolution, product entitlement, or customer objection should not be split in ways that remove necessary context. The framework uses semantic chunking aligned with business objects: account summaries, opportunity stages, ticket narratives, case-resolution steps, meeting decisions, product-capability descriptions, and contractual obligations. Each chunk receives embeddings, sparse index terms, temporal metadata, entity links, sensitivity labels, and relationship edges to related objects. This allows retrieval to combine semantic similarity with structured business constraints.

The fifth step is retrieval orchestration. The proposed system uses hybrid retrieval: sparse keyword retrieval for exact terms, dense semantic retrieval for conceptual similarity, metadata filtering for account and role constraints, graph traversal for relationship expansion, and reranking for evidence prioritization. For example, a query asking "Why is the Acme renewal at risk?" should retrieve recent support escalations, unresolved severity-one cases, contract renewal notes, executive meeting transcripts, and product adoption

signals associated with the account. It should not retrieve generic churn articles unless the user asks for general guidance. Retrieval must therefore be customer-scoped, temporally aware, permission-aware, and decision-task aware.

The sixth step is grounded generation. The language model receives only the retrieved and authorized context, task instructions, user role, output schema, and safety constraints. The generation layer is instructed to distinguish evidence-based statements from inference, to cite source passages, to abstain when evidence is insufficient, and to surface uncertainty. For high-risk tasks, the generator should produce a structured response containing answer, supporting evidence, assumptions, confidence indicator, unresolved evidence gaps, recommended next action, and review requirement. The goal is not merely to produce fluent answers but to produce decision-support artifacts that can be inspected and challenged.

The seventh step is verification and feedback. The framework includes automated checks for source grounding, answer relevance, citation coverage, policy violations, and contradiction between generated statements and retrieved evidence. User feedback is captured at the level of answer usefulness, evidence relevance, missing context, incorrect inference, and business outcome. This feedback updates retrieval rules, prompt templates, chunking strategies, and knowledge-quality indicators. The evaluation loop is aligned with decision-intelligence thinking: AI outputs must be linked to governance and lifecycle improvement rather than treated as static model responses.

The methodology also incorporates explicit trust controls derived from AI risk-management frameworks. Controls include role-based access enforcement before retrieval, personally identifiable information masking when appropriate, customer-data residency restrictions, prompt-injection detection, retrieval-source allowlists, audit logs, source attribution, answer abstention, red-team testing, and escalation rules for sensitive outputs. These controls are treated as architectural components, not compliance afterthoughts. A CRM RAG system becomes trustworthy only when technical design, governance policy, user training, and operational monitoring function as an integrated system.

The proposed framework further distinguishes between internal and external outputs. Internal outputs support employees in understanding customer context and planning actions. External outputs, such as customer emails, renewal statements, service responses, or contractual explanations, require higher review thresholds. In the architecture, external-facing drafts must pass citation checks, policy filters, and human approval before release. This distinction is crucial because the same generated insight may be acceptable as an internal hypothesis but unacceptable as a customer-facing claim without verification.

Finally, the methodology defines implementation as an iterative maturity model. At Level 1, enterprises deploy retrieval over curated knowledge bases for limited support questions. At Level 2, CRM records and customer documents are indexed with metadata and role-based access. At Level 3, account-specific synthesis, opportunity intelligence, and service-resolution assistance are enabled. At Level 4, feedback loops, automated evaluation, and governance dashboards support continuous improvement. At Level 5, the architecture supports multi-agent workflow orchestration with strict human oversight for high-risk customer decisions. This staged approach reduces implementation risk and allows organizations to build trust progressively.

4. System Architecture or Conceptual Model

The proposed architecture consists of eight layers: source systems, ingestion and governance, knowledge representation, retrieval services, orchestration and prompt management, generation and verification, user interaction, and monitoring. Each layer is independently scalable but governed by shared identity, metadata, and policy services. The architecture is designed for cloud-native deployment using microservices, event-driven ingestion, distributed indexing, vector search, graph storage, model gateways, and observability pipelines. The system can be integrated with commercial CRM platforms, data warehouses, contact-center tools, document repositories, and enterprise identity providers.

The source-system layer includes structured CRM entities such as accounts, contacts, leads, opportunities, activities, products, cases, entitlements, renewals, quotes, and contracts. It also includes unstructured sources such as meeting notes, call transcripts, chat logs, email interactions, customer-success plans, implementation documents, support knowledge articles, product release notes, and sales playbooks. External sources may include public company filings, news, partner documentation, or industry reports, but these must be clearly separated from internal customer records. Source classification is necessary because internal facts, external signals, and generic playbooks have different evidentiary strength.

The ingestion layer performs extraction, transformation, classification, and policy attachment. Structured CRM data is ingested through APIs, change-data-capture streams, or scheduled batch pipelines. Unstructured documents are processed through parsers, speech-to-text systems, email connectors, and document loaders. Each ingested object is assigned metadata, including source system, object type, customer entity, timestamp, owner, confidentiality level, retention period, and access policy. Policy is attached at ingestion time to prevent unauthorized retrieval later. The ingestion layer also detects duplicates, stale records, missing customer identifiers, and conflicting versions of documents.

The knowledge-representation layer stores customer intelligence in complementary forms. A relational or lakehouse store preserves normalized structured records. A vector index stores embeddings for semantic retrieval. A sparse index supports exact keyword and identifier search. A knowledge graph stores relationships among accounts, contacts, opportunities, cases, products, contracts, and documents. This multi-representation design is necessary because CRM questions often combine semantic, structured, and relational reasoning. For example, "Which unresolved onboarding issues may affect the renewal?" requires linking tickets, implementation milestones, account hierarchy, product usage, and renewal dates.

The retrieval layer implements a query-planning pipeline. User queries are classified by intent, risk level, customer entity, and task type. Query expansion identifies synonyms, product names, abbreviations, and CRM-specific terms. Policy filters restrict retrieval to records the user is authorized to access. Hybrid retrieval combines vector similarity, keyword matching, metadata filters, and graph expansion. Reranking prioritizes passages based on relevance, recency, source reliability, customer specificity, and decision usefulness. Retrieval results are then packaged into evidence bundles with source identifiers, snippets, confidence scores, and metadata.

The orchestration layer coordinates prompts, tools, memory, and response schemas. It selects a prompt template based on task type, such as account briefing, support response, churn explanation, opportunity summary, objection handling, or executive escalation. It controls whether the model may call additional tools, such as CRM APIs, analytics services, or policy databases. It also enforces output structure. For example, an account-briefing template may require sections on recent interactions, open risks, stakeholder map, active opportunities, product usage signals, recommended questions, and cited evidence. A service-resolution template may require issue summary, likely cause, relevant policy, recommended response, and escalation path.

The generation layer uses an enterprise LLM gateway rather than direct model access. The gateway supports model selection, cost control, prompt logging, content filtering, token budgeting, and latency management. The generator is instructed to rely only on retrieved evidence for customer-specific claims. It must explicitly state when evidence is insufficient. It must not invent customer names, contract clauses, product commitments, or support outcomes. For sensitive use cases, the model produces a draft that requires human approval. For lower-risk internal summarization, the system may allow direct display but still logs evidence and user feedback.

The verification layer is responsible for post-generation checks. It validates that each key claim is supported by at least one retrieved source, detects contradictions between answer and evidence, checks whether restricted information appears in the output, and ensures that citations correspond to authorized documents. It also detects prompt-injection attempts embedded in retrieved documents, such as malicious instructions in customer emails or uploaded files. Outputs failing verification are blocked, rewritten, or returned with an abstention message. This layer is central to trusted customer intelligence because it separates generation from acceptance.

The user-interaction layer provides role-specific interfaces. Sales users may interact through account pages, opportunity workspaces, meeting-preparation panels, or mobile CRM assistants. Service agents may use case-resolution copilots embedded in ticket queues. Customer-success managers may use health-score narratives, renewal-risk summaries, and adoption-insight views. Executives may use aggregated customer intelligence dashboards with drill-down evidence. In each interface, the assistant should display evidence sources, confidence indicators, and feedback controls. Users should be able to mark evidence as irrelevant, report missing context, request more sources, or convert generated recommendations into CRM tasks.

The monitoring layer captures technical, trust, and business metrics. Technical metrics include retrieval latency, generation latency, index freshness, token cost, cache hit rate, and system availability. Trust metrics include groundedness, citation coverage, hallucination rate, policy-violation rate, abstention rate, and human override rate. Business metrics include time saved, case-resolution improvement, opportunity-stage acceleration, renewal-risk detection, knowledge reuse, and user adoption. These metrics are visualized in governance dashboards for AI product owners, CRM administrators, compliance teams, and business leaders.

Scalability is addressed through distributed indexing, asynchronous ingestion, caching, and workload partitioning. Frequently accessed knowledge articles and sales playbooks can be cached, while customer-specific records require strict freshness. Vector indexes can be partitioned by tenant, region, customer segment, or data domain. Event-driven ingestion updates indexes when new tickets, notes, or transcripts arrive. High-volume call transcripts can be summarized hierarchically before indexing. The architecture also supports model routing: smaller models can handle classification and query rewriting, while larger models handle synthesis and reasoning-intensive tasks.

Security and privacy are embedded across all layers. Identity and access management determine what the user may retrieve. Data-loss-prevention policies detect sensitive content. Encryption protects data at rest and in transit. Audit trails record user query, retrieved sources, generated output, model version, prompt template, and feedback. Retention policies determine when indexed chunks expire. The architecture also supports customer-specific isolation, which is especially important for multi-tenant SaaS CRM deployments. Trustworthy customer intelligence requires that retrieval be correct not only semantically but also legally and organizationally.

The conceptual model can be summarized as a closed-loop system: customer interactions generate knowledge; knowledge is governed, indexed, and connected; user queries trigger policy-aware retrieval; retrieved evidence conditions generation; verification checks the output; users act on or correct the recommendation; feedback improves retrieval and knowledge quality. This loop transforms CRM from a passive record system into an adaptive knowledge-management environment. The architecture does not replace CRM workflows; it augments them with evidence-grounded intelligence embedded in the point of decision.

5. Evaluation Criteria / Performance Metrics

Evaluation of enterprise CRM RAG requires a multi-dimensional framework because a system can be technically fluent but operationally unsafe, or retrieval-accurate but unusable in sales workflows. The proposed evaluation framework includes retrieval quality, generation quality, trust and compliance, decision usefulness, scalability, and adoption. Each metric should be measured by task type and risk tier, since acceptable performance differs between low-risk note summarization and high-risk contract interpretation. Evaluation should combine offline benchmarks, expert review, synthetic test sets, red-team scenarios, and production telemetry.

Retrieval quality measures whether the system retrieves the right evidence. Core metrics include precision at k, recall at k, mean reciprocal rank, normalized discounted cumulative gain, metadata-filter accuracy, source freshness, and entity-resolution accuracy. In CRM settings, retrieval must be evaluated not only by textual relevance but also by customer specificity. A retrieved document about a similar customer should not be treated as equivalent to a document about the target account. Evaluation sets should include realistic queries such as “summarize open blockers for the renewal,” “what objections did the customer raise last quarter,” and “which cases indicate dissatisfaction with onboarding.”

Generation quality measures whether the answer is useful, coherent, complete, and grounded. Standard language-quality metrics are insufficient because enterprise CRM answers must be supported by retrieved evidence. Therefore, the evaluation framework includes answer faithfulness, citation correctness, evidence coverage, contradiction rate, abstention appropriateness, and unsupported-claim rate. RAGAS-style reference-free evaluation is useful because it decomposes RAG assessment into context relevance, faithfulness, answer relevance, and related dimensions, enabling faster evaluation cycles when labeled ground truth is limited [22].

Trust and compliance metrics measure whether the assistant respects enterprise constraints. These include access-control violation rate, sensitive-data leakage rate, prompt-injection susceptibility, policy-filter precision, audit completeness, privacy-redaction accuracy, and human-review compliance for external outputs. For regulated industries, additional metrics may include data-residency compliance, retention enforcement, and evidence preservation. Trust metrics should be reviewed by compliance, legal, security, and business stakeholders because CRM intelligence may involve contractual obligations, customer confidentiality, and regulated communications.

Decision usefulness measures whether the output improves human work. Metrics include time-to-account-briefing, reduction in manual search effort, first-contact resolution improvement, escalation-preparation time, opportunity-review quality, renewal-risk detection, and user-rated usefulness. It is also important to measure negative outcomes, such as overreliance, ignored evidence, unnecessary escalations, or excessive abstentions. Decision usefulness should be evaluated through controlled user studies, A/B testing,

and workflow telemetry rather than only offline model metrics. In customer operations, a correct answer that arrives too late or fails to fit workflow constraints may have limited value.

Scalability metrics measure whether the architecture can support enterprise usage. These include indexing throughput, query latency, retrieval latency, generation latency, concurrent-user capacity, cost per answer, cache efficiency, index-update delay, and degradation under peak load. CRM systems often experience workload spikes around quarter-end sales cycles, product incidents, renewal deadlines, and campaign launches. The system must maintain reliability when many users request account summaries, opportunity analyses, or support guidance simultaneously. Latency targets should be task-specific: service-agent use cases may require near-real-time response, while executive synthesis may tolerate longer processing.

Reliability metrics should include model-version stability, retrieval drift, prompt-regression rate, and output variability across repeated runs. Machine learning evaluation research in software defect prediction demonstrates the importance of comparing models empirically rather than assuming one algorithm is universally superior [13]. The same principle applies to CRM RAG: retrieval models, embedding strategies, chunk sizes, rerankers, prompts, and generation models should be benchmarked against realistic enterprise tasks. Evaluation must be continuous because customer knowledge, product documentation, and user behavior change over time.

Predictive and diagnostic evaluation can also benefit from fault-prediction thinking. Gunda's work comparing Random Forest, Logistic Regression, and K-Neighbors for fault prediction underscores that model effectiveness varies by data characteristics, target definition, and evaluation criteria [19]. In CRM RAG, this suggests that retrieval and generation pipelines should be tested across diverse customer segments, data densities, and interaction histories. A system that performs well for large enterprise accounts with rich records may perform poorly for small accounts with sparse notes. Evaluation should therefore stratify results by account size, region, product line, knowledge-source type, and user role.

A robust evaluation program should include adversarial testing. Test cases should cover ambiguous customer names, conflicting documents, stale contracts, missing permissions, irrelevant retrieved passages, malicious instructions in documents, sensitive data, and questions requiring abstention. The assistant should be rewarded not only for answering but also for refusing unsupported claims. For high-risk use cases, expert reviewers should grade evidence adequacy and business appropriateness. The most important evaluation question is not "Can the model answer?" but "Can the system produce a reliable, authorized, evidence-supported answer that improves the decision at hand?"

6. Results and Discussion / Analytical Discussion

The analytical evaluation of the proposed framework indicates that enterprise CRM RAG should be understood as a knowledge-governance architecture rather than a chatbot pattern. The primary value is not the ability to generate natural language; it is the ability to retrieve, structure, validate, and explain customer knowledge at the moment of decision. Compared with conventional CRM search, the architecture supports natural-language questions, semantic matching, cross-source synthesis, and role-specific output. Compared with conventional dashboards, it produces narrative explanations supported by evidence. Compared with standalone LLMs, it reduces hallucination risk by grounding outputs in authorized customer knowledge.

The first expected benefit is improved knowledge accessibility. CRM users often spend substantial time locating prior interactions, interpreting notes, checking support cases, and verifying product or contract context. A RAG system can reduce this friction by retrieving relevant evidence and producing concise summaries. For example, before a renewal meeting, a customer-success manager may ask for the top unresolved risks and receive a response grounded in recent tickets, product usage notes, executive emails, and contract milestones. The assistant does not merely summarize the account; it identifies evidence-backed risk themes and points to the underlying sources.

The second expected benefit is improved decision quality. Sales and service decisions are often made under time pressure and incomplete information. By consolidating customer evidence, the architecture can support better opportunity qualification, escalation prioritization, churn-risk interpretation, and objection handling. However, the system must distinguish between evidence and inference. If the assistant infers that a customer may be dissatisfied because several onboarding tickets remain unresolved, it should label this as an inference and cite the tickets. This improves managerial accountability because users can inspect the basis of the recommendation rather than accept an opaque score or narrative.

The third expected benefit is improved knowledge reuse. CRM organizations often recreate knowledge repeatedly because insights from one team are not accessible to another. Support teams may know recurring product issues; sales teams may know customer objections; customer-success teams may know adoption barriers; product teams may know roadmap constraints. The proposed architecture can connect these knowledge domains through shared indexing, entity linking, and permission-aware retrieval. It can transform isolated notes and transcripts into reusable knowledge assets while preserving access rules and source provenance.

The fourth expected benefit is reduced hallucination and unsupported claims. RAG does not eliminate hallucination, but it changes the control surface. Instead of relying on a model's internal memory, the architecture can measure whether retrieved evidence supports generated claims. This creates opportunities for automated verification, citation checks, and abstention. In CRM, abstention is not a failure when evidence is insufficient; it is often the safest and most truthful response. A trusted assistant should say "I do not have evidence for that claim in the available customer records" rather than produce a plausible but unsupported answer.

The fifth expected benefit is stronger compliance posture. Because retrieval is policy-aware, the system can enforce role-based access before sensitive content reaches the model. This is essential when CRM data includes personal information, contractual terms, pricing details, health or financial information, or confidential customer communications. The architecture can also produce audit records showing which sources were retrieved, which model was used, what answer was generated, and whether a human approved the output. These audit trails are critical for enterprise governance, especially when generated content influences customer communication.

The sixth expected benefit is workflow integration. CRM users are unlikely to adopt an assistant that requires them to leave their workflow, manually paste context, or interpret generic output. The architecture is therefore designed for embedded use: account pages, case queues, opportunity reviews, call-preparation screens, and executive dashboards. The assistant should retrieve context automatically from the current CRM object while allowing users to refine the query. Embedded evidence display is also important because users must be able to trust and challenge the answer without conducting a separate search.

The framework also reveals several trade-offs. Increasing retrieval scope may improve recall but reduce precision and increase latency. More context may improve evidence coverage but can overload the model or trigger long-context weaknesses. Strict access controls improve compliance but may frustrate users if important evidence is hidden without explanation. More aggressive abstention improves safety but may reduce perceived usefulness. Higher-quality reranking improves answer relevance but increases cost and complexity. Enterprise architecture must therefore tune these trade-offs by task type and risk tier rather than apply one universal configuration.

Another important issue is source quality. RAG systems are only as reliable as the knowledge they retrieve. CRM notes may be incomplete, biased, outdated, or written in shorthand. Support tickets may contain preliminary diagnoses later corrected elsewhere. Sales notes may reflect subjective impressions rather than verified facts. Contracts may have amendments that supersede earlier clauses. Therefore, the architecture includes source reliability indicators, recency weighting, contradiction detection, and version control. The assistant should not treat all retrieved text as equally authoritative. A signed contract, updated entitlement policy, and informal meeting note should have different evidentiary weights.

The analytical discussion also highlights the importance of feedback loops. User corrections should not disappear into logs. If users repeatedly mark certain sources as irrelevant, the retrieval pipeline should adjust ranking or metadata. If the assistant frequently misses a particular knowledge base, ingestion should be reviewed. If generated summaries omit important customer context, prompt templates should be revised. If users override recommendations in predictable ways, business rules or retrieval logic may need adjustment. Continuous learning in this architecture does not necessarily mean fine-tuning the LLM; it often means improving data quality, retrieval design, prompts, policies, and workflow integration.

The proposed architecture can support multiple CRM use cases. In sales intelligence, it can generate account briefs, identify buying-committee signals, summarize objections, and prepare discovery questions. In customer service, it can retrieve relevant policies, summarize case histories, suggest resolution paths, and draft responses. In customer success, it can explain health-score drivers, identify adoption barriers, and prepare renewal plans. In marketing, it can synthesize campaign feedback and segment-specific objections. In

executive decision support, it can aggregate evidence-backed themes across strategic accounts. Each use case should have a tailored prompt schema, retrieval profile, and evaluation benchmark.

A notable architectural implication is that CRM RAG should combine vector retrieval with knowledge graphs. Vector search is effective for semantic similarity, but customer intelligence often depends on relationships: which contacts belong to which account, which opportunities link to which products, which cases involve which entitlements, and which contracts govern which regions. A graph layer can expand retrieval from a customer entity to related objects while respecting access policies. This reduces the risk of retrieving semantically similar but organizationally irrelevant documents. It also allows the assistant to reason over customer hierarchies, stakeholder relationships, and account histories.

The results of the analytical framework suggest that enterprise RAG maturity depends on organizational readiness as much as model capability. Organizations with poor CRM hygiene, inconsistent documentation, weak access controls, and limited process ownership will struggle to deploy trusted RAG. Conversely, organizations with strong metadata, disciplined knowledge management, clear governance, and measurable workflows can obtain significant value even with conservative model configurations. The architecture therefore encourages enterprises to treat RAG deployment as a knowledge-management modernization program rather than a plug-in AI feature.

The framework also identifies risks. Users may overtrust fluent answers, especially when responses include citations that appear authoritative. Retrieved evidence may be selectively presented, leading to biased narratives. Sensitive information may be exposed through misconfigured permissions. Prompt injection may manipulate the generator through malicious text in retrieved documents. Evaluation metrics may overemphasize answer relevance while undermeasuring business risk. To mitigate these risks, the architecture includes human review for high-risk outputs, policy filters, adversarial testing, evidence inspection, and governance dashboards.

From a research perspective, the proposed architecture contributes to the intersection of information systems, CRM, knowledge management, and AI engineering. It extends traditional CRM frameworks by adding evidence-grounded generative interfaces and trust-oriented retrieval governance. It extends RAG research by situating retrieval and generation within enterprise access-control, workflow, and decision-accountability constraints. It extends knowledge-management research by showing how generative interfaces can operationalize organizational knowledge while preserving provenance and human oversight. The result is a conceptual foundation for future empirical studies and enterprise prototypes.

7. Future Research Directions

Future research should empirically evaluate the proposed architecture in real CRM environments across industries such as healthcare, financial services, enterprise software, telecommunications, retail, and manufacturing. Such studies should measure not only answer accuracy but also workflow outcomes, including sales-cycle efficiency, case-resolution time, renewal-risk detection, customer satisfaction, and employee trust. Longitudinal studies are especially important because user behavior, knowledge quality, and organizational reliance on AI evolve over time.

A second direction is the development of CRM-specific RAG benchmarks. Existing RAG benchmarks often focus on open-domain question answering or general knowledge tasks. CRM requires benchmarks involving account histories, service tickets, opportunity notes, contract clauses, customer hierarchies, and role-based permissions. Benchmark datasets should include ambiguous queries, conflicting evidence, stale documents, sensitive data, and required abstention cases. Privacy-preserving synthetic CRM datasets may be useful for academic research because real CRM data is difficult to share.

A third direction is adaptive retrieval governance. Future systems should dynamically adjust retrieval strategy based on user role, task risk, evidence availability, query ambiguity, and customer-data sensitivity. Low-risk internal summarization may use broader retrieval, while high-risk customer-facing recommendations may require stricter source authority and human approval. Adaptive retrieval could reduce cost and latency while maintaining trust. Research is needed to determine how such systems should decide when to retrieve, how much to retrieve, and when to abstain.

A fourth direction is integration between RAG and causal customer intelligence. Current RAG systems are strong at retrieving and summarizing evidence but weaker at distinguishing correlation from causation. In customer operations, decision-makers often need

causal explanations: whether churn risk is driven by product defects, pricing, onboarding delay, competitor activity, or stakeholder turnover. Future research could combine RAG with causal graphs, event-sequence modeling, and counterfactual reasoning to produce more rigorous decision support.

A fifth direction is human-AI collaboration design. Research should examine how different evidence displays, confidence indicators, feedback mechanisms, and review workflows influence user trust and decision quality. Overly cautious systems may be ignored, while overly confident systems may promote automation bias. The optimal interface may vary by role: executives may need concise narratives with drill-down evidence, while service agents may need step-by-step resolution guidance. Human factors research is therefore essential for trusted CRM intelligence.

A sixth direction is governance automation. As RAG systems scale, manual review of prompts, retrieval configurations, and knowledge sources becomes difficult. Future research should explore automated policy testing, retrieval red-teaming, drift detection, citation-quality monitoring, and compliance dashboards. These tools could help organizations maintain trust as CRM knowledge evolves. Governance automation should remain interpretable and auditable because enterprise accountability cannot be delegated entirely to opaque monitoring systems.

Finally, future research should examine multi-agent CRM architectures. Multiple specialized agents may handle account research, support diagnosis, contract retrieval, meeting preparation, and executive summarization. However, multi-agent systems introduce coordination, accountability, and security challenges. Future work should define protocols for agent communication, evidence sharing, conflict resolution, human approval, and auditability. A trusted CRM intelligence platform may ultimately combine RAG, agents, knowledge graphs, analytics, and human governance into a unified decision-support ecosystem.

8. Conclusion

This paper proposed a scalable enterprise architecture for retrieval-augmented generation in trusted CRM knowledge management. The central argument is that enterprise customer intelligence requires more than generative fluency. It requires governed retrieval, source attribution, access control, evidence verification, human oversight, continuous evaluation, and workflow integration. CRM data is fragmented, dynamic, sensitive, and decision-relevant; therefore, direct use of large language models without enterprise grounding is insufficient and potentially risky. he proposed framework integrates CRM source systems, ingestion governance, semantic and graph-based knowledge representation, hybrid retrieval, prompt orchestration, grounded generation, verification, user feedback, and monitoring. It supports sales intelligence, customer service, customer success, marketing analysis, and executive decision support while preserving trust requirements. The architecture treats RAG as a socio-technical knowledge-management system rather than a narrow model enhancement technique.

The evaluation framework emphasizes retrieval quality, generation faithfulness, trust and compliance, decision usefulness, scalability, reliability, and adoption. This multi-dimensional perspective is essential because CRM RAG systems must be judged by their ability to improve human decisions safely, not merely by their ability to produce fluent answers. The analytical discussion shows that the greatest value of RAG lies in transforming CRM from a passive repository into an adaptive, evidence-grounded intelligence environment. Future research should validate the framework empirically, develop CRM-specific benchmarks, explore adaptive retrieval, integrate causal reasoning, study human-AI collaboration, automate governance, and extend the architecture toward multi-agent customer-intelligence systems. As enterprises increasingly deploy generative AI in customer-facing functions, trusted RAG architectures will become critical infrastructure for responsible and scalable customer knowledge management.

References

[1] A. Payne and P. Frow, "A Strategic Framework for Customer Relationship Management," *Journal of Marketing*, vol. 69, no. 4, pp. 167–176, 2005, doi: 10.1509/jmkg.2005.69.4.167.

[2] Gunda, S. K., Yettapu, S. D. R., Bodakunti, S., & Bikki, S. B. (2023). Decision Intelligence Methodology for AI-Driven Agile Software Lifecycle Governance and Architecture-Centered Project Management. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *4*(1), 102-108. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I1P112

[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," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020. Available: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[4] M. Alavi and D. E. Leidner, "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," *MIS Quarterly*, vol. 25, no. 1, pp. 107–136, 2001, doi: 10.2307/3250961.
[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.
[6] E. W. T. Ngai, L. Xiu, and D. C. K. Chau, "Application of Data Mining Techniques in Customer Relationship Management: A Literature Review and Classification," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2592–2602, 2009, doi: 10.1016/j.eswa.2008.02.021.
[7] Gunda, S. K. G. (2023). The Future of Software Development and the Expanding Role of ML Models. *International Journal of Emerging Research in Engineering and Technology*, *4*(2), 126-129. https://doi.org/10.63282/3050-922X.IJERET-V4I2P113
[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017. Available: https://proceedings.neurips.cc/paper/7181-attention-is-all-you-need
[9] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 6769–6781, doi: 10.18653/v1/2020.emnlp-main.550.
[10] National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, Jan. 2023, doi: 10.6028/NIST.AI.100-1. Available: https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf
[11] H. Chen, R. H. L. Chiang, and V. C. Storey, "Business Intelligence and Analytics: From Big Data to Big Impact," *MIS Quarterly*, vol. 36, no. 4, pp. 1165–1188, 2012, doi: 10.2307/41703503.
[12] 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, Article 248, pp. 1–38, 2023, doi: 10.1145/3571730.
[13] Siswantoro, M. Z. F. N., & Yuhana, U. L. (2023). *Software defect prediction based on optimized machine learning models: A comparative study*. Teknika, 12(2), 166–172. https://doi.org/10.34148/teknika.v12i2.634
[14] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, doi: 10.2307/25148625.
[15] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval Augmentation Reduces Hallucination in Conversation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 3784–3803, doi: 10.18653/v1/2021.findings-emnlp.320.
[16] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2312.10997, 2023. Available: https://arxiv.org/abs/2312.10997
[17] T. B. Brown et al., "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020. Available: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html
[18] A. R. Zablah, D. N. Bellenger, and W. J. Johnston, "Customer Relationship Management Implementation Gaps," *Journal of Personal Selling & Sales Management*, vol. 24, no. 4, pp. 279–295, 2004, doi: 10.1080/08853134.2004.10749038.
[19] Borandag, E. (2023). Software fault prediction using an RNN-based deep learning approach and ensemble machine learning techniques. *Applied Sciences, 13*(3), Article 1639. https://doi.org/10.3390/app13031639
[20] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, "When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada, 2023, pp. 9802–9822, doi: 10.18653/v1/2023.acl-long.546.
[21] High-Level Expert Group on Artificial Intelligence, *Ethics Guidelines for Trustworthy AI*, European Commission, Brussels, Belgium, 2019. Available: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai
[22] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated Evaluation of Retrieval Augmented Generation," arXiv preprint arXiv:2309.15217, 2023. Available: https://arxiv.org/abs/2309.15217
[23] T. H. Davenport and L. Prusak, *Working Knowledge: How Organizations Manage What They Know*. Boston, MA, USA: Harvard Business School Press, 1998.
[24] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the Middle: How Language Models Use Long Contexts," arXiv preprint arXiv:2307.03172, 2023. Available: https://arxiv.org/abs/2307.03172
[25] OpenAI, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023. Available: https://arxiv.org/abs/2303.08774