Contextual Knowledge Retrieval Systems for Industrial Operations: Efficiency Quantification and Maritime Application

I. Foundational Architectures for Contextual Knowledge Retrieval

The optimization of operational efficiency in high-value industrial sectors, such as engineering, aviation, and maritime maintenance, mandates a radical shift in how technical documentation is managed and searched. Traditional information retrieval systems, reliant on keyword matching, have proven inadequate for managing the massive volume of complex, unstructured data characteristic of these domains.

1.1 The Shift from Lexical to Contextual Search

Traditional keyword, or lexical, search methods operate by matching explicit terms in a query to identical or lexically similar terms within documents. This paradigm breaks down when dealing with vast industrial documentation—including thousands of proprietary PDFs, maintenance logs, and engineering specifications—due primarily to the issues of synonymy and the difficulty in grasping conceptual relationships across disparate sources.[1] A significant consequence of this inefficiency is that knowledge workers, such as engineers and technicians, are estimated to spend up to 30% of their time simply searching for and consolidating necessary information across various unstructured documents.[2] This non-productive time directly contributes to increased operational costs and prolonged Mean Time To Resolution (MTTR) for critical faults.

Semantic search systems overcome these limitations by utilizing vector embeddings to capture the inherent *meaning* or *context* of both the query and the document content. Instead of searching by exact word, the system maps queries into a high-dimensional vector space, allowing it to find and return existing documents or content chunks that are semantically relevant, even if they do not contain the exact phraseology used by the technician.[3] This ability to grasp intent rather than just vocabulary is essential for domain-specific troubleshooting where technical terminology varies widely.

1.2 Pure Semantic Search versus Retrieval-Augmented Generation

The selection of a contextual search architecture depends critically on the desired output: simple document discovery and verification, or synthesized, prescriptive action.

Pure Semantic Search (Document Discovery)

Pure semantic search focuses solely on the retrieval component. The user query is vectorized, and the system retrieves the most relevant content chunks (nearest neighbors) from a vector database based on conceptual meaning. This approach is optimal when the primary requirement is content retrieval for verification or compliance evidence.

For environments handling highly sensitive data—such as proprietary engineering blueprints or maritime defense information—this approach offers a key advantage in data privacy and compliance.[3] By utilizing local embeddings and indexing technologies (e.g., an on-premises vector database), user queries and the retrieved document content remain isolated from external, public-facing services. This is a crucial governance consideration, as it mitigates the potential for data leakage that can occur when sensitive context is transmitted to external large language model (LLM) Application Programming Interfaces (APIs).

Retrieval-Augmented Generation (RAG) (Knowledge Synthesis)

RAG represents a more complex, compound system that integrates the semantic retrieval capability with a generative model.[4] It is necessary when the application requires moving beyond merely finding a document to delivering a contextual, synthesized, or *prescriptive* answer. The RAG pipeline involves three distinct steps: first, retrieving relevant documents using semantic search; second, augmenting the user's query by inserting the retrieved context as grounding information; and finally, generating a new, coherent, synthesized response using the LLM.[5]

This synthesis capability is particularly valuable in industrial maintenance. For instance, in the manufacturing sector, RAG frameworks are essential for delivering prescriptive maintenance recommendations.[7] Instead of simply locating the relevant section in a manual, the RAG system generates a structured, step-by-step action plan for a specific fault, combining information from multiple sources (manuals, sensor data analysis, and historical fix logs) into a single, actionable output.[3]

The choice of architecture is therefore fundamentally a governance and functional decision. While pure semantic search prioritizes security and document verification, RAG provides the necessary conversational and actionable synthesis required for modern troubleshooting.

Table 1: Comparative Analysis of Pure Semantic Search and RAG

Feature	Pure Semantic Search (Vector Retrieval)	Retrieval-Augmented Generation (RAG)
Primary Output	Existing source documents, content chunks.	Synthesized, generated, conversational response.
Complexity	Lower (Retrieval only).	Higher (Retrieval + Generation).
Data Privacy	High; allows for local embeddings and processing.[3]	Lower; often sends context to external LLM APIs (risk of data export).[3]
Use Case Focus	Document discovery, content verification, compliance evidence.	Conversational AI, prescriptive actions, synthesized troubleshooting.[3]

II. Quantification of Efficiency Gains in Industrial Operations

The justification for implementing contextual search systems relies on demonstrable efficiency gains, primarily measured by the reduction in non-productive time and the accelerated resolution of technical issues.

2.1 The Business Imperative: Measuring Reduced Lookup Time

The primary operational metric targeted by these systems is the Mean Time To Resolution (MTTR), which quantifies the average duration from the reporting of an issue to its final resolution.[8] A high MTTR often indicates slow information lookup, inefficient escalation, and reliance on inadequate knowledge bases. Studies demonstrate that improvements in MTTR correlate directly with customer satisfaction and retention. For example, a 10% improvement in MTTR can result in a 1% increase in customer retention and a 5% increase in customer satisfaction.[10] The following case studies illustrate the quantified impact of RAG and semantic indexing across various industrial domains.

2.2 Case Study 1: Accelerated Field Service Ticket Resolution via RAG

Field service operations, particularly in large sectors like telecom and utilities, face constant challenges due to the geographical dispersion of assets and the complexity of modern equipment. Field technicians often rely heavily on 'tribal knowledge' or inefficient manual search processes.[11]

Global telecom and utility providers have deployed RAG-powered knowledge assistants directly to field service teams. These systems uniquely integrate disparate knowledge sources—including structured historical support tickets (such as those from JIRA), unstructured developer discussions from internal platforms, and official proprietary maintenance logs—into a unified, semantically searchable repository.[12] The utility of RAG in this context lies in its capacity to link cross-functional data; linking a developer's comment about a known bug directly to the field service manual's troubleshooting section breaks down siloed knowledge. This holistic integration of information is reflected in non-maintenance metrics, with implementation timelines for enterprise clients reportedly shortened by 11%.[12]

The quantifiable reduction in lookup and resolution time is substantial:

- **Time Savings:** Organizations reported that the average time taken to resolve a service ticket was reduced by **20% to 50%** following RAG implementation.[14]
- **Operational Capacity:** For one global provider, the system saved over **15,000 agent**

hours within the first six months of deployment.[12] This freed support staff to focus on complex queries rather than routine lookups.
- **Revenue Impact:** The operational improvement, driven by faster first-time resolution and reduced churn, contributed directly to a **3.8% boost in customer retention**, resulting in significant annualized revenue uplift.[12]

Example Scenario: A field technician encounters an intermittent network fault at a remote location. Instead of contacting a supervisor or manually searching dozens of large PDFs, the technician queries the AI: *"Known causes for intermittent power cycling on the B6-series router when ambient temperature exceeds 40°C?"* The RAG system synthesizes the answer by referencing a 2-year-old JIRA ticket confirming a firmware issue, a developer forum comment detailing a hotfix workaround, and the specific cooling requirements from the component manual.

2.3 Case Study 2: High-Speed Semantic Extraction in Unstructured Aviation Data

The aviation sector is characterized by regulatory strictness and an overwhelming volume of complex, unstructured documentation, including airworthiness certificates, compliance reports, and vast technical manuals.[2] Managing this data effectively is critical but challenging, often leading to time-consuming and laborious search processes for mechanics.[1]

Utilizing LLM-based systems and advanced semantic indexing, organizations are transforming previously labor-intensive workflows, particularly document processing and compliance tracking. These technologies enable rapid data extraction and consolidation across disparate formats, turning unstructured data into actionable insights.[2]

The efficiency gains are demonstrated not only in operational speed but also in implementation cycles:

- **Document Processing Speed:** Companies have achieved up to **91% time savings** in automated document processing when shifting from traditional Machine Learning (ML) approaches to LLM-based systems.[15]
- **Implementation Acceleration:** This shift reduces the typical training and implementation cycle for document processing systems from months to just one or two days.[15]
- **Comprehension Improvement:** Precursor systems utilizing ontology-driven semantic search showed that workers could more intuitively grasp complex relationships among content, leading to a reduction in the overall time required to search for information.[1]

Example Scenario: An airline compliance officer receives a new airworthiness directive requiring all fleet manuals to be checked for a specific material safety warning related to a composite structure sealant. Keyword search is inadequate because the sealant may be referred to by proprietary names or chemical formulas. Semantic search instantaneously identifies all relevant sections across dozens of technical manuals across 50 aircraft, ensuring rapid, auditable compliance.

2.4 Case Study 3: Heavy Industrial Engineering and Prognosis

Modern industrial maintenance requires timely, context-aware intervention driven by real-time sensor data. The goal is to shift operations management from reactive firefighting to proactive, automated problem-solving.[11] This capability is achieved through advanced frameworks that integrate data types previously treated separately.

The Prescriptive Agents based on RAG for Automated Maintenance (PARAM) framework illustrates this integration by moving beyond anomaly detection to deliver intelligent, context-aware, prescriptive maintenance recommendations.[7] Traditional anomaly detection relies on specialized ML models analyzing structured numerical data, such as bearing vibration frequencies. These models identify the fault but require subsequent manual cross-referencing with text-based manuals to determine the required action.

The RAG architecture addresses this bottleneck:

- **Detection Accuracy:** Prior work demonstrated that LLMs, by serializing numerical sensor data into natural language prompts, achieved anomaly detection accuracy comparable to, and in some cases, superior to, conventional numerical ML models.[7] This validated the core premise that LLMs can accurately classify faults based on telemetry.
- **Efficiency Gain in Prescriptive Action:** PARAM leverages RAG to synthesize maintenance manuals, domain expertise, and real-time knowledge following fault classification. This automates the critical cross-referencing step, instantly generating a structured action plan.[7] While specific MTTR reductions are emerging, the integration eliminates the lag between fault identification and action plan generation, offering significant efficiency improvements over manual documentation lookup, potentially reflecting the **91% time savings** observed in similar document processing tasks.[15]

Example Scenario: A sensor monitoring a main turbine bearing flags high vibration frequencies (structured data). The RAG system links this specific numerical signature to an inner race fault classification, immediately retrieves the manufacturer's exact prescriptive maintenance procedure for that fault, and generates a time-stamped work order with detailed component handling instructions, all within seconds.

Table 2: Quantified Efficiency Metrics: Lookup Time Reduction in Industrial Use Cases

Use Case Domain	Metric Targeted	Legacy Performance (Baseline)	AI/RAG Performance (Target)	Reduction/Gain
Field Service / Support	Average Resolution Time	High reliance on manual search and escalation.	Reduced MTTR by 20–50%.	Significant efficiency boost.[14]
Field Service / Support	Agent Hours Saved	Excessive hours spent on knowledge search/consolidation.	Over 15,000 hours saved in 6 months (telecom).	Massive operational capacity gain.[12]
Document Processing	Data Extraction Time	Training ML for document processing (6 months).	LLM-based IDP/OCR (max 2 days implementation).	91% time savings on implementation cycle.[15]
Aviation Maintenance	Knowledge Worker Search Time	Up to 30% of time spent searching for data.	Time minimized through intuitive retrieval.	Significant operational capacity increase.[2]

III. Strategic Application in the Maritime Sector

The shipbuilding and maritime industries are characterized by the strategic importance of proprietary data, fragmented supply chains, and complex regulatory compliance.[16] Onboard a vessel, the challenge is compounded by the critical nature of maintenance—errors translate directly to safety risks, regulatory non-compliance, and severe financial losses due to unplanned downtime. Immediate access to precise, verified technical information is paramount.

Module Title: Immediate Fault Code Resolution via Contextual Retrieval

Narrative: A key application of contextual retrieval systems in the maritime environment is the immediate and accurate resolution of proprietary engine or equipment fault codes. Modern vessels rely on complex programmable logic controllers (PLCs) and engine monitoring systems that frequently generate cryptic error messages, such as Mercury Marine Code 173 ("Fuel System. Fuel Pressure is high," resulting in "Engine Power Limited").[17] An engineer needs to bypass manual index navigation and instantly retrieve the exact troubleshooting page, safety warnings, specific repair steps, and component diagrams associated with that code. A RAG system designed for this purpose must ensure a high degree of precision in retrieval and synthesis.

Bullets (RAG Architecture for Vessel Deployment):

- **Data Ingestion and Indexing:** The system must ingest highly fragmented, proprietary Original Equipment Manufacturer (OEM) manuals (e.g., Wärtsilä, MAN), shipyard construction specifications, and classification society requirements. These documents must be rigorously segmented, or *chunked*, based on semantic relevance. Crucially, a domain-specific ontology must be modeled to map core entities, such as specific fault codes (e.g., ME-486-FP-H), component identifiers, and safety procedures, thereby enriching the vector index and facilitating accurate retrieval.[1]
- **Query-Artifact Matching:** When a query is entered—such as the example "What is error E-486?"—the system must execute a precise semantic search. This ensures high **Retrieval Accuracy** [18], instantly associating the query with the correct component manual and section. The system must meet a stringent **Reciprocal Rank** requirement, meaning the first result returned must be the correct, actionable document or synthesized answer.[18]
- **Context Augmentation and Generation:** The LLM receives the prompt alongside the retrieved, grounded context chunks (e.g., the fault code definition, the identified cause, and the mandated repair procedure). The LLM's role is to generate a synthesized, prioritized, and easy-to-read action plan, explicitly incorporating all associated safety warnings and cross-referencing data from the ship’s computerized maintenance management system (CMMS).

Example Scenario: “What is error E-486?” → Exact Page/Action Plan

- **Query:** *"What is the prescribed action plan and safety warning for Main Engine PLC fault* *code ME-486-FP-H, indicating high fuel pressure?"*

- **System Response (Synthesized):** The system first validates the query and identifies the relevant source content, stating the context is retrieved from the **Wärtsilä Manual, Section 4.5.2, page 92**.[19]
  - **Fault Identified:** Code ME-486-FP-H (High Fuel Pressure, Non-Critical - Service Engine Soon [17]).
  - **Cause:** Suspected failure of the Fuel Rail Pressure Sensor (FRPS-1).
  - **Prescriptive Action Plan (Step 1 of 5): SAFETY WARNING:** Before proceeding, immediately close the main fuel bunker valve and vent residual pressure from the rail using tool T-211-FP. Failure to vent pressure may result in severe injury.[17]
  - **Integrated Check:** The system also cross-references the code with the CMMS to check if FRPS-1 is due for replacement, providing a holistic maintenance recommendation.

For safety-critical tasks, the system must employ a high confidence threshold. If the RAG system cannot achieve a high confidence score for synthesizing a complete and accurate, verbatim answer (e.g., a specific torque setting or safety step), it is engineered to revert to its pure semantic search mode, returning only the source document.[19] This prioritization of accuracy over synthesis prevents a critical failure known as an "Incomplete Answer" or "Not Extracted" failure.[20]

IV. System Limitations, Failure Analysis, and Governance

While RAG systems provide massive efficiency gains, they are **compound systems** [4] where failure can occur in the retrieval component, the ranking component, or the final generation stage.[20] A thorough understanding of these failure points is essential for robust, mission-critical deployment.

4.1 The Retrieval Component: Source of Index Errors and False Positives

Retrieval accuracy is the foundation of RAG. If the initial search phase fails, the subsequent generative model will struggle, regardless of its sophistication.[20]

- **Missing Content:** If critical proprietary data—such as a specific engine update bulletin or a recent shipyard trial report—was never properly ingested, processed, or indexed into the vector database, the RAG system will be unable to find it. This results in the system confidently responding that the information does not exist, leading to operational delays or inaccurate actions.[20]
- **Missing Top Ranked Documents:** Index errors or poorly tuned retrieval parameters can cause relevant documents to be indexed but ranked too low in the search results list. Consequently, the information is not passed to the LLM as context (the 'context window'), leading to a generated answer that is suboptimal or incorrect.[20]
  - **Mitigation Requirement:** To counteract poor ranking, specialized RAG evaluation metrics must be employed. These include **Recall@K**, which measures the proportion of all relevant documents that appear within the top K retrieved results, and **Reciprocal Rank**, which ensures the most relevant document appears as the first result.[18] Consistent index validation and data preprocessing pipelines are also critical, as the effort required to preprocess domain knowledge and store it in the vector database is non-trivial.[6]

4.2 The Generation Component: Hallucination and Misinformation Amplification

The most significant risk of RAG implementation is its potential to become a **Hallucination Amplifier**.[21]

- **The Hallucination Amplifier Effect:** RAG is commonly adopted under the assumption that it eliminates hallucinations. However, if the underlying retrieval index contains inaccurate, biased, or outdated internal enterprise data, the LLM will confidently integrate these falsehoods into its synthesized answer, citing the source and creating an illusion of factual credibility.[21] For instance, retrieving an obsolete regulatory compliance procedure or a superseded engine maintenance protocol could lead to severe consequences, component failure, or regulatory penalties.
- **Inconsistent Retrieval and RAG Overload:** Retrieval failure can also trigger generation problems. If the semantic search component cannot find *any* relevant documents due to a failed search or poor indexing, the LLM may revert to relying solely on its internal, generalized training data (parametric knowledge) to answer the domain-specific query. This leads to traditional hallucination—an incorrect but often confident answer.[21] Conversely, if retrieval brings *too much* non-specific data (a failure of granularity configuration [4]), the LLM can suffer from **RAG Overload**, misinterpreting key facts or introducing contradictions when attempting to summarize the excessively large context window.[21]

4.3 Governance and Evaluation Requirements

For engineering and maintenance applications, standard evaluation metrics designed for general LLMs (such as BLEU or ROUGE) are inadequate because they fail to assess the quality of the retrieval component.[18] A robust deployment requires dedicated governance and continuous RAG-specific evaluation.

Blindly trusting retrieval-based AI systems without rigorous source verification constitutes a fundamental governance failure.[21] Organizations must mandate formal processes to ensure the currency and accuracy of all indexed documents, especially technical specifications and regulatory content.

Required RAG-Specific Metrics for Robustness [18]:

- **Retrieval Accuracy:** The percentage of queries for which the retrieved set successfully contained the correct supporting documents.
- **Context Recall:** A measurement of whether all necessary information required to fully answer the query was present within the retrieved context chunks passed to the LLM, independent of their precise rank position.

These metrics focus on ensuring that the *grounding* information provided to the generative model is both complete and accurate, thereby limiting the opportunity for generation failure.

Table 3: Common RAG Failure Modes and Mitigation Strategies

Failure Mode	**Description / Risk**	**Impact in Maritime/Engineering**	**Mitigation Strategy**
Hallucination Amplifier	Retrieval of outdated or incorrect internal data, which the LLM presents as verified fact.	Catastrophic operational failure due to obsolete repair procedures or flawed safety advice.	Rigorous source verification, explicit time-stamping of all indexed documents, and external validation checks.

Missing Content	Relevant documents were not processed or indexed into the vector database.	Operational delays; system reports "no answer found" when a critical procedure exists.	Continuous monitoring of the data ingestion pipeline; full coverage analysis of all proprietary documentation.
Incorrect Specificity	Retrieval occurs at the wrong document chunk granularity (too broad or too narrow).	Inefficient RAG Overload or an incomplete answer is generated due to ambiguous context.	Granularity configuration tuning [4]; use of nDCG metrics to evaluate ranking quality.[18]
Low Context Recall	Not all necessary context needed for a full answer is retrieved, despite finding some relevant documents.	Generates a partially correct but non-actionable or non-compliant answer.	Explicit RAG evaluation focusing on Context Recall metrics; careful adjustment of chunk size and overlap.[18]

V. Conclusion and Strategic Recommendations

5.1 Summary of Foundational Success and Demonstrated ROI

Contextual search technologies, particularly those leveraging the Retrieval-Augmented Generation framework, represent a paradigm shift in industrial knowledge management. They provide compelling, quantifiable return on investment by systematically reducing non-productive search time—which historically consumed up to 30% of a knowledge worker's operational capacity [2]—and by accelerating Mean Time to Resolution by **20% to 50%** in field service environments.[14] The observed gains translate directly into massive operational capacity increases (e.g., saving over 15,000 agent hours [12]) and accelerated project timelines.

5.2 Strategic Roadmap for Secure RAG Implementation

For mission-critical sectors like maritime and aviation, strategic adoption must prioritize data trust and retrieval integrity over generative capability. The following phased approach is recommended for secure deployment:

1. **Phase I: Data Trust and Indexing Infrastructure:** Capital investment must initially focus on the infrastructure required for high-quality data ingestion and cleansing, rather than solely on selecting the largest LLM. Implement robust preprocessing pipelines and vector indexing to chunk and tag proprietary manuals, logs, and specifications. Establish stringent governance rules requiring continuous index validation and source verification to ensure data currency and accuracy.
2. **Phase II: Security and Compliance Checkpoint:** Determine the acceptable privacy risk profile. For highly sensitive maritime operations involving defense or proprietary engineering data, mandate a secure, on-premises **Pure Semantic Search** model as the foundational layer.[3] This ensures user queries and context remain local, adhering to rigorous compliance standards before external APIs are considered.
3. **Phase III: Pilot RAG Deployment and Evaluation:** Introduce RAG for generative synthesis only after achieving and validating highly accurate retrieval in the secured environment. Pilot deployments must be rigorously evaluated using RAG-specific metrics, including Recall@K, Reciprocal Rank, and Context Recall.[18] These metrics isolate retrieval performance, ensuring that the first result provided to the operator is consistently the correct, actionable answer.
4. **Phase IV: Operationalization and Feedback Loops:** Integrate the verified RAG system directly into key maintenance workflows, such as CMMS and digital work order systems. Implement automated feedback loops, essential for compound systems [4], to continuously monitor the LLM's outputs and adjust the indexing and ranking parameters based on user-reported inaccuracies or search failures.

5.3 Final Assessment: The Trust Mandate

The ability to instantly resolve a critical fault, such as identifying the prescribed action for engine code ME-486-FP-H, offers tangible safety and financial benefits to a vessel. However, the successful implementation of contextual search systems in engineering environments hinges not on the intelligence of the LLM, but on the trustworthiness and currency of the underlying knowledge index. Failures in RAG are overwhelmingly *data failures*.[20] Therefore, the endeavor must be treated as the creation of a secure, continuously monitored, and highly adaptable data asset, where governance and infrastructure integrity are prioritized above all else.

Works cited

1. Ontology-Driven Visualization System for Semantic Search | Request PDF - ResearchGate, accessed October 17, 2025, https://www.researchgate.net/publication/224238447_Ontology-Driven_Visualization_System_for_Semantic_Search
2. How AI Solves the Challenges of Data Extraction from Unstructured Aviation Documents, accessed October 17, 2025, https://www.eplaneai.com/blog/how-ai-solves-the-challenges-of-data-extraction-from-unstructured-aviation-documents
3. RAG Vs Semantic Search: Key Differences Explained, accessed October 17, 2025, https://customgpt.ai/rag-vs-semantic-search/
4. RAGOps: Operating and Managing Retrieval-Augmented Generation Pipelines - arXiv, accessed October 17, 2025, https://arxiv.org/html/2506.03401v1
5. What is RAG? - Retrieval-Augmented Generation AI Explained - AWS - Updated 2025, accessed October 17, 2025, https://aws.amazon.com/what-is/retrieval-augmented-generation/
6. Seven Failure Points When Engineering a Retrieval Augmented Generation System - arXiv, accessed October 17, 2025, https://arxiv.org/html/2401.05856v1
7. Prescriptive Agents based on RAG for Automated Maintenance (PARAM) - arXiv, accessed October 17, 2025, https://arxiv.org/html/2508.04714v2
8. How Generative AI Can Help Reduce Resolution Time - Workativ, accessed October 17, 2025, https://workativ.com/ai-agent/blog/generativei-ai-reduce-resolution-time
9. Time to Resolution - Push.ai, accessed October 17, 2025, https://www.push.ai/metric/time-to-resolution
10. Time to Resolution: What Is It, and How to Reduce It - Aidbase - AI, accessed October 17, 2025, https://www.aidbase.ai/blog/time-to-resolution
11. A Maintenance Revolution: Reducing Downtime With AI Tools, accessed October 17, 2025, https://sloanreview.mit.edu/article/a-maintenance-revolution-reducing-downtime-with-ai-tools/
12. 5 Metrics to Evaluate the ROI of Enterprise Knowledge Assistants - Signity Solutions, accessed October 17, 2025, https://www.signitysolutions.com/blog/metrics-to-evaluate-the-roi-of-enterprise-knowledge-assistants
13. [2510.08667] RAG4Tickets: AI-Powered Ticket Resolution via Retrieval-Augmented Generation on JIRA and GitHub Data - arXiv, accessed October 17, 2025, https://arxiv.org/abs/2510.08667
14. Enhancing Customer Support Efficiency & Satisfaction with RAG - Squirro, accessed October 17, 2025, https://squirro.com/squirro-blog/future-of-customer-support-with-llms-a-breakt

hrough-for-efficiency-satisfaction
15. Data Extraction 2025: OCR vs IDP vs LLM - Which Technology to Choose? - PaperOffice, accessed October 17, 2025, https://start.paperoffice.com/en/data-extraction-documents-llm-vs-machine-learning
16. Navigating shipbuilding 4.0: analysis and classification of technologies for the digital transformation of the sector - TU Delft Repository, accessed October 17, 2025, https://repository.tudelft.nl/file/File_08bed111-6281-4af1-909a-cbef88d522ef
17. MERCURY Fault Codes - Boat, Yacht, Jet Ski & Marine Engine Manual PDF - Scribd, accessed October 17, 2025, https://www.scribd.com/document/621372283/MERCURY-Fault-Codes-Boat-Yacht-Jet-Ski-Marine-Engine-Manual-PDF
18. RAG evaluation: a technical guide to measuring retrieval-augmented generation - Toloka, accessed October 17, 2025, https://toloka.ai/blog/rag-evaluation-a-technical-guide-to-measuring-retrieval-augmented-generation/
19. Quickstart: Semantic Ranking - Azure AI Search - Microsoft Learn, accessed October 17, 2025, https://learn.microsoft.com/en-us/azure/search/search-get-started-semantic
20. Seven Ways Your RAG System Could be Failing and How to Fix Them - Label Studio, accessed October 17, 2025, https://labelstud.io/blog/seven-ways-your-rag-system-could-be-failing-and-how-to-fix-them/
21. Is Your AI Lying to You? The Hidden Risks of RAG and How to Fix Them - BI Group Australia, accessed October 17, 2025, https://bigroup.com.au/is-your-ai-lying-to-you-the-hidden-risks-of-rag-and-how-to-fix-them/