Chunking Strategies for Retrieval in Enterprise Documents: A Comparative Study and the QI-RAG Proposal

Maryane de Castro Lima[1,2], **Thomaz Maia de Almeida**[1,2]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

[2]Núcleo de visão computacional e engenharia (NUVEN)

maryane.castro@sti.ufc.br, thomaz.maia@ifce.edu.br

Abstract. *Chunking is a critical yet underexplored step in enterprise RAG pipelines. We compare four strategies (character-based, recursive, semantic, hierarchical) on 500 queries from financial documents (VDR-IBM REAL-MM-RAG) using BGE-M3 and Nomic embeddings with LLM-as-a-judge metrics. We also propose Question-Indexed RAG (QI-RAG), where a Gemma 3 LLM generates diverse questions per chunk for vector indexing, with threshold-based hybrid retrieval. Baselines show clear progression (53.6% $\rightarrow$ 75.5% contextual precision). QI-RAG reaches 83.1% precision and 0.82 accuracy, cutting failures to 2.4% (14.3$\times$ below character-based).*

1. Introduction

Retrieval-Augmented Generation (RAG) grounds large language models (LLMs) in external evidence, mitigating hallucinations by conditioning generation on retrieved passages [Lewis et al. 2020, Gao et al. 2024]. RAG is especially relevant in enterprise environments, where knowledge is distributed across heterogeneous, frequently changing documents — policies, contracts, financial reports, FAQs, technical manuals — making parametric memorization (knowledge stored directly in model weights) insufficient [Bruckhaus 2024].

Modern RAG pipelines rely on dense retrieval: documents are chunked, embedded as vectors, and matched against the query by similarity [Karpukhin et al. 2020]. Despite advances in embeddings and retrieval algorithms [Sarthi et al. 2024, Asai et al. 2024], chunking — the atomic retrieval unit — remains largely heuristic, even though its boundaries determine what is retrieved, how much irrelevant context enters, and the quality of the generated response.

This problem is amplified in enterprise documents: numerical tables, multi-column text, hierarchical lists, and cross-page references mean naive splits fragment tables, separate definitions from exceptions, and scatter narratives across non-adjacent chunks. A further *semantic gap* — between colloquial user queries (e.g., "what was the company's annual expenditure?") and technical, formal document language — impairs similarity-based search even with robust embeddings.

The contributions are: (i) an empirical comparative analysis of four chunking strategies on 500 queries from financial documents, with progression 53.6% $\rightarrow$ 75.5% in contextual precision; (ii) **QI-RAG**, a question-based indexing approach with hybrid retrieval, achieving 83.1% precision and 0.82 overall accuracy.

2. Related Work

Chunking strategies. Wang [Wang et al. 2024, Wang et al. 2025] (EMNLP 2024, ACL 2025) shows document segmentation is a first-order RAG variable: oversized chunks add irrelevant content while undersized ones lose coherence — both, however, evaluate on open-domain QA, leaving open the transfer to table-heavy enterprise reports. Bennani and Moslonka [Bennani and Moslonka 2026] show a non-linear context-length effect (a "context cliff" past $\sim$2.5k tokens) and no measurable overlap benefit, but on Natural Questions with sparse retrieval, not the dense embeddings and complex documents evaluated here. Qu [Qu et al. 2025] (NAACL 2025) questions whether semantic chunking justifies its cost — motivating this work's index-transformation alternative. Taiwo and Yusoff [Taiwo and Yusoff 2026] reach a convergent conclusion in a different vertical (oil and gas), still at the chunk-boundary level. Cunha [Cunha et al. 2025] targets another institutional vertical (hospital regulatory documents), but optimizes reranking and embedding choice rather than chunking boundaries. Toresan [Toresan et al. 2025] refines single-pass semantic chunking with linguistic pre-segmentation (Dual Semantic Chunker), improving boundaries but leaving the same gap: what gets indexed is still the raw chunk. RAPTOR [Sarthi et al. 2024] (ICLR 2024), Self-RAG [Asai et al. 2024] (ICLR 2024), and HybridRAG [Sarmah et al. 2024] extend RAG beyond simple chunking — via summary hierarchies, adaptive retrieval fine-tuning, and knowledge-graph hybridization, respectively — but require substantial resources or architectural change beyond this work's local-deployment scope; HybridRAG hybridizes at the retrieval-source level (vector store plus graph), unlike QI-RAG's index-level hybridization (questions plus chunk fallback).

Retrieval via question generation: the basis of QI-RAG. HyDE [Gao et al. 2023] (ACL 2023) generates a hypothetical answer document and embeds it for retrieval, sidestepping direct query-document comparison — at the cost of an LLM call per query and no reusable index. Raina and Gales [Raina and Gales 2024] instead transform the *index*: chunks are decomposed into atomic units with up to 15 candidate questions each, pruned by redundancy; their evaluation, however, is restricted to single-hop SQuAD (Wikipedia), with no fallback when matching fails and no table/figure treatment. Eponon [Eponon et al. 2025] adds per-paper summaries and a two-stage lexical-then-semantic reranker, reaching Recall@3 of 0.84 (+60% over chunking) on single-hop NLP-paper retrieval (plus separate multihop-QA experiments), but — like Raina and Gales — without enterprise documents, table/figure handling, or a fallback mechanism. Lee [Lee 2025] decomposes multihop queries into sub-questions before question-question matching, the closest antecedent to QI-RAG, yet stays confined to Wikipedia-derived benchmarks with the same two gaps. QI-RAG builds on this index-transformation direction — instead of transforming the *query* as HyDE does, it transforms the *index* by replacing chunks with questions likely to be asked about them. It distinguishes itself from all four by targeting the financial enterprise domain, adding type-specific treatment for tables and figures, using a content-driven adaptive question count instead of a fixed cap-and-prune policy, and pairing the index with threshold-based hybrid retrieval and chunk-level fallback.

Models and evaluation. The Gemma 3 family [Google DeepMind 2025] (Google DeepMind), adopted here, offers instruction-tuned 4B–27B models with a 128K-token

context window and strong parameter efficiency for local GPU deployment. Embedding models BGE-M3, introduced by Chen [Chen et al. 2024] (ACL 2024) with dense, sparse, and multi-vector retrieval support, and Nomic Embed v1.5 [Nussbaum et al. 2024] (nomic-ai), both open-source and state-of-the-art on MTEB, are evaluated as a comparative pair to isolate chunking-strategy impact from embedding choice. The RAGAS framework [Es et al. 2024] (EACL 2024) decomposes retrieval quality into contextual metrics; the LLM-as-a-judge paradigm [Zheng et al. 2023] (NeurIPS 2023) provides reference-free generation evaluation; both run via DeepEval [Confident AI 2024].

3. Methodology

This section describes the dataset used, the document preprocessing pipeline, the four chunking strategies evaluated, and the QI-RAG proposal, along with the language models, indexing configuration, and evaluation protocol adopted.

3.1. Dataset

We evaluate the strategies on the *VDR-IBM REAL-MM-RAG Financial Report* dataset [Racine AI 2024] (500 non-null-query samples, SEED=42), a public benchmark of IBM financial report pages pairing a PNG page with a natural language query and reference answer. The reports exhibit typical enterprise complexity — numerical tables, multi-column text, embedded charts, cross-page references — suitable for evaluating RAG with rich layout.

3.2. Document Preprocessing

Since the pipeline operates in text-only mode, PNG images are converted to Markdown via Docling [IBM Research 2024], which extracts structured text, tables, and layout information. Tables are preserved in Markdown syntax (`|col|col|`); non-transcribable figures are marked `<!-- image -->`. The resulting Markdown constitutes the retrieval corpus for all strategies.

3.3. Chunking Strategies Evaluated

We evaluate four baseline strategies, all implemented via LangChain (a Python library for retrieval and generation pipeline orchestration[1]). For the configurable-size strategies (character-based, recursive, semantic), we use 800-character chunks with 200-character overlap, keeping these hyperparameters fixed to isolate the effect of each structural strategy:

- **Character-based** (`CharacterTextSplitter`): splits at fixed character boundaries, ignoring document structure; may fragment sentences, table rows, or list items. The simplest baseline.
- **Recursive** (`RecursiveCharacterTextSplitter`): splits using priority-ordered separators (paragraph breaks, newlines, sentence boundaries, whitespace), falling back to finer granularity only past the target size. Tends to preserve titles, paragraphs, and lists as atomic units.
- **Semantic** (`SemanticChunker`): groups consecutive sentences by embedding similarity into chunks with high internal coherence, with the similarity threshold calibrated by percentile over consecutive-sentence cosine distances.

[1] https://github.com/langchain-ai/langchain

- **Hierarchical** (`MarkdownHeaderTextSplitter`): leverages the Markdown heading hierarchy from Docling (H1, H2, H3) to align chunks with the document's logical structure, retaining source-section metadata (title and level) after splitting. Especially suitable for well-structured enterprise documents such as financial reports.

For a fair comparison, table header preservation is applied uniformly across all strategies: when a table is fragmented, its original header row is replicated in each derived segment so any retrieved chunk stays independently interpretable — the only content-type-specific intervention shared across all methods.

3.4. Proposed Approach: Question-Indexed RAG (QI-RAG)

QI-RAG is motivated by the semantic gap between the colloquial language of user queries and the technical-formal language of enterprise documents. Instead of indexing chunks directly, QI-RAG indexes *LLM-generated questions* about each chunk, bringing the search space closer to real user queries. The approach operates in four phases, illustrated in Figure 1.

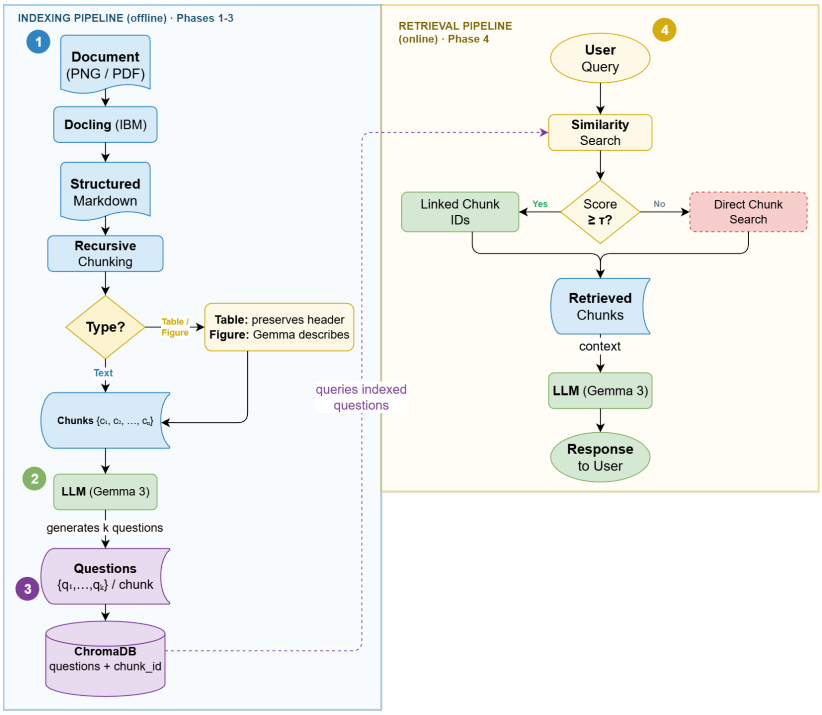


Figure 1. QI-RAG pipeline. Indexing (Phases 1–3): chunking with table/figure detection, per-chunk question generation, and question indexing in ChromaDB. Retrieval (Phase 4): similarity search over indexed questions, with direct fallback to chunks when no match clears the threshold.

3.4.1. Phase 1 – Chunking and Type Detection

QI-RAG builds on recursive chunking (`RecursiveCharacterTextSplitter`, 800 characters, 200 overlap) and classifies each resulting chunk c_i into one of three types based on Markdown syntax patterns: **textual** (default), **tabular** (presence of `|---|` row sepa-

rator), or **figure** (`<!-- image -->` marker inserted by Docling for non-transcribable images).

For **tabular** chunks, when a large table spans multiple segments, the original header row is replicated atop each derived segment so any retrieved slice stays interpretable — e.g., "14.1" is meaningless without its header "Revenue (USD B)". For **figure** chunks, Gemma 3 generates a textual description from the surrounding context and section metadata (title, preceding paragraph); this becomes an **additional chunk** c_{desc}, inserted after the figure marker and following the same question-generation and indexing flow as textual chunks, with the original marker chunk retained alongside it.

3.4.2. Phase 2 – Question Generation

For each chunk $c_i \in C$, Gemma 3 generates $k_i \geq k_{\min}{=}5$ semantically diverse questions $Q_i = \{q_1^i, \ldots, q_{k_i}^i\}$. The generation prompt instructs the LLM to vary question types: definitions ("What does X refer to?"), numerical values ("What was the value of X in period Y?"), comparisons ("How did X change between Y and Z?"), and process descriptions ("How is X calculated?"). The prompt also adjusts by chunk type: tabular chunks prioritize quantitative and comparative questions about the table's columns; figure description chunks focus on trends, highlighted values, and visual relationships described in the text. Full prompt templates are released for reproducibility.[2]

3.4.3. Phase 3 – Question Indexing

Each question q_j^i is vectorized by embedding function ϕ (BGE-M3 or Nomic Embed v1.5) and stored in a dedicated ChromaDB collection with a pointer to the identifier of its source chunk c_i:

$$\text{index} \leftarrow \left\{ (\phi(q_j^i),\ \text{id}(c_i)) \;\middle|\; \forall\, i,\ \forall\, j \right\} \tag{1}$$

This separates the *search index* (questions, colloquial language) from the *answer repository* (chunks, technical language): populated with question-like sentences, the index's vector space sits much closer to the user query distribution than raw document text.

3.4.4. Phase 4 – Hybrid Retrieval with Fallback

Given user query u, $\phi(u)$ is compared against the question collection by cosine similarity. If the top-ranked question reaches similarity $\geq \tau = 0.75$, the $n = 4$ most similar questions are retrieved and their linked source chunks returned as context (primary path); $\tau = 0.75$ was chosen empirically to balance precision (avoiding weak matches) against recall (fallback only when the index lacks a strong match).

When no question reaches τ, a second search runs directly over the chunk collection, returning the n closest chunks by cosine similarity (fallback path). This hybrid design keeps even vocabulary-distant queries in context, consistent with the failure-rate drop from 6.1% (hierarchical) to 2.4% (QI-RAG). Retrieved chunks from either path are concatenated and passed to Gemma 3 for final response generation.

[2] `https://github.com/maryane-castro/eniac-prompts-qirag`

3.5. Language Models

For response generation and, in QI-RAG, per-chunk question generation, we use **Gemma 3** [Google DeepMind 2025] (Google DeepMind) in its **4B**, **12B**, and **27B** instruction-tuned variants, reflecting resource-constrained local deployment. Baselines (character, recursive, semantic, hierarchical) are evaluated with Gemma 27B as a high-quality generation ceiling; QI-RAG is evaluated across all three sizes to test whether its retrieval gains hold at reduced parameter counts.

3.6. Indexing and Retrieval Configuration

All chunks from each strategy are indexed in ChromaDB with two open-source embedding models: **BGE-M3** [Chen et al. 2024] (BAAI/bge-m3) and **Nomic Embed v1.5** [Nussbaum et al. 2024] (nomic-ai/nomic-embed-text-v1.5). For QI-RAG, the same pair vectorizes generated questions, creating additional ChromaDB collections. Retrieval uses cosine similarity, returning $n = 4$ nearest candidates in all scenarios.

For QI-RAG, the question-generation prompt (Section 3.4) instructs the LLM to produce *at least* $k_{\min} = 5$ semantically diverse questions per chunk, one per question type from Phase 2 (definitions, numerical values, comparisons, process descriptions), with no fixed upper bound: the model generates further questions when the chunk covers more distinct facts, rather than truncating dense chunks or padding short ones. In practice, tabular and figure chunks yield 5–9 questions, while long textual chunks may exceed this range. The similarity threshold is set to $\tau = 0.75$: similarities below this value trigger the direct-chunk fallback. Experiments ran on an Intel Core i7 machine with an NVIDIA GeForce RTX 5060 Ti GPU (16 GB VRAM).

3.7. Evaluation Protocol

Evaluation is conducted at two levels using DeepEval [Confident AI 2024] with the LLM-as-a-judge paradigm [Zheng et al. 2023] and the RAGAS taxonomy [Es et al. 2024]:

Level 1 — Retrieval. RAGAS contextual metrics: **Contextual Precision** (relevant chunks ranked above irrelevant ones), **Contextual Recall** (coverage of reference information), and **Contextual Relevance** (fraction of statements pertinent to the query). We also report the **Failure Rate** (proportion of queries with zero precision). For QI-RAG with Gemma 27B, we additionally report **Overall Accuracy**: the proportion of queries for which the top-ranked retrieved chunk was judged, by the same GEval judge (GPT-OSS 120B), to contain the reference answer's information; not measured for the baselines.

Level 2 — Generation. The Gemma 3 response is evaluated via DeepEval [Confident AI 2024] with: **GEval – Correctness** (LLM-as-a-judge with chain-of-thought, GPT-OSS 120B via Groq API as judge, score 0–1); **Answer Relevance** (statements pertinent to the query); **BERTScore** (semantic similarity with reference via BERT); **METEOR** (n-gram overlap with flexible alignment); and **ParaScore** (cosine similarity between sentence-transformer embeddings of generated and reference responses).

4. Results

4.1. Comparison of Chunking Strategies

Table 1 reports contextual retrieval metrics for all evaluated configurations (4 strategies $\times$ 2 embedding models, 500 queries each), ordered by ascending contextual precision.

Table 1. Average contextual metrics per configuration (500 queries).

Configuration	Ctx. Precision	Ctx. Recall	Ctx. Relevance
Character + BGE-M3	0.458	0.478	0.345
Character + Nomic	0.615	0.671	0.343
Recursive + BGE-M3	0.718	0.683	0.329
Recursive + Nomic	0.722	0.634	0.336
Semantic + BGE-M3	0.731	0.688	0.338
Semantic + Nomic	0.743	0.701	0.344
Hierarchical + BGE-M3	0.748	0.712	0.341
Hierarchical + Nomic	0.762	0.724	0.349

Figure 2 shows a consistent progression from worst to best: Character $<$ Recursive $<$ Semantic $<$ Hierarchical. Hierarchical chunking achieves the highest baseline precision (0.755), surpassing character-based by 40.9% — highlighting the benefit of aligning segment boundaries with document structure.

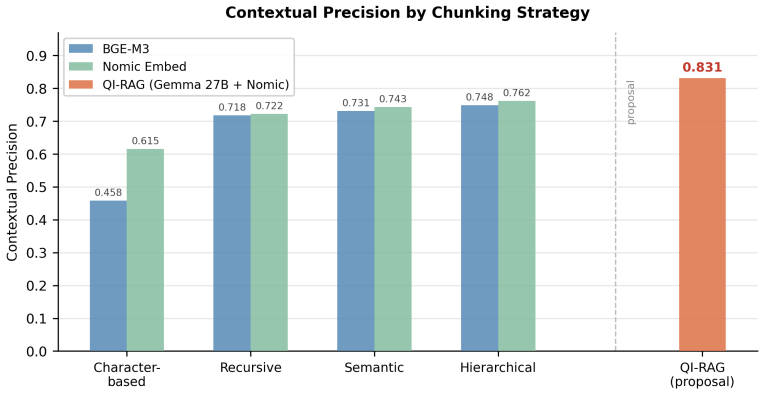


Figure 2. Contextual precision by chunking strategy and embedding model (500 queries). The orange bar represents QI-RAG with Gemma 27B and Nomic Embed.

Precision gains confirm that structural awareness produces more coherent segments and concentrates relevant evidence at the top of the retrieval list. The jump from character-based to recursive (+57% relative for BGE-M3) is the largest single step in the progression and is explained by separator-aware splitting: paragraph and newline boundaries prevent mid-sentence or mid-table fragmentation that systematically degrades character-based retrieval. The smaller gap between recursive and semantic (+1.8%) suggests that sentence-level grouping by embedding similarity adds modest additional coherence beyond paragraph-level splitting in this dataset.

Contextual relevance remains stable across strategies ($\approx$0.33–0.35), shaped primarily by domain-query alignment rather than chunk boundaries. Notably, character-based chunking with BGE-M3 shows slightly higher contextual relevance (0.345) than

recursive (0.329): smaller, more homogeneous chunks contain less off-topic content per segment — at the cost of much lower global precision and recall.

Both embedding models (BGE-M3 and Nomic) produce comparable results, with Nomic outperforming BGE-M3 by up to 34.3% under character-based chunking (0.615 vs. 0.458), a gap that shrinks to under 2% for hierarchical (0.762 vs. 0.748). The large gap at character-based level reflects BGE-M3's sensitivity to fragmented input: when chunk boundaries split tables or sentences, the dense multi-vector representation loses coherence more sharply than Nomic's matryoshka embeddings. In the recursive strategy, Nomic exhibits lower recall than BGE-M3 (0.634 vs. 0.683) despite higher precision, suggesting a precision/recall trade-off that attenuates for strategies with better-defined boundaries. Within the evaluated configurations, chunking strategy had a larger impact than embedding choice on system performance.

Figure 2's orange bar previews this trajectory: QI-RAG's 0.831 sits visibly above the entire baseline cluster, foreshadowing the retrieval gains detailed in Section 4.3.

4.2. Failure Rate Analysis

Figure 3 shows the retrieval failure rate — queries with zero contextual precision — for each strategy. Character-based chunking leaves more than one third of queries without any relevant retrieved chunk. Hierarchical reduces this to 6.1%, but the largest leap is QI-RAG's 2.4% — a 14.3× reduction versus the weakest baseline. The threshold-based fallback is designed for exactly this scenario: when the primary question-indexed search lacks sufficient confidence, a second direct-chunk search targets the residual failures.

The failure-rate progression mirrors the precision pattern in Figure 2: the largest reduction occurs at the character-to-recursive step (34.4% to 9.3%, a 73% relative drop), while gains from recursive through semantic to hierarchical are comparatively marginal (9.3% to 7.8% to 6.1%). This reinforces that separator-aware splitting, not fine-grained semantic or hierarchical refinement, is what most reliably prevents complete retrieval failure; QI-RAG's fallback then addresses the residual cases that structural chunking alone cannot resolve.

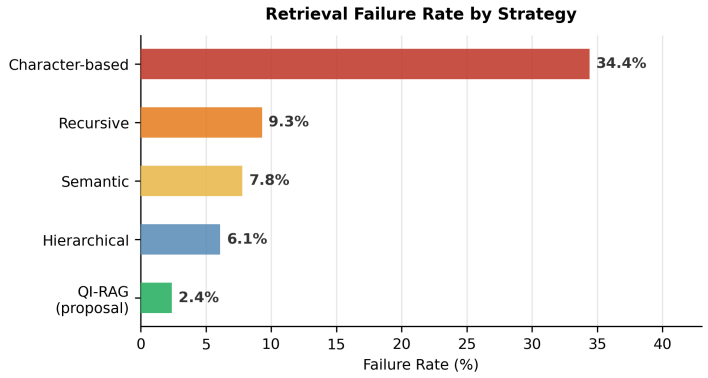


Figure 3. Retrieval failure rate by strategy (500 queries, average over BGE-M3 and Nomic). QI-RAG reduces failures by 14.3× vs. character-based.

4.3. QI-RAG Evaluation

Tables 2 and 3 compare QI-RAG against the best baseline (hierarchical) on retrieval and generation metrics for the three Gemma 3 model sizes evaluated with Nomic Embed (best

individual configuration).

Table 2. QI-RAG vs. best baseline — retrieval metrics (Nomic Embed, 500 queries).

Configuration	Prec.	Rec.	Rel.
Hierarchical (Nomic)	0.762	0.724	0.349
QI-RAG + Gemma 4B	0.798	0.763	0.354
QI-RAG + Gemma 12B	0.817	0.779	0.359
QI-RAG + Gemma 27B	**0.831**	**0.793**	**0.366**

Table 3. QI-RAG vs. best baseline — generation metrics (Nomic Embed, 500 queries).

Configuration	Ans. Rel.	METEOR	BERT	Para
Hierarchical (Nomic)	0.65	0.37	0.64	0.50
QI-RAG + Gemma 4B	0.61	0.34	0.61	0.46
QI-RAG + Gemma 12B	0.65	0.38	0.65	0.51
QI-RAG + Gemma 27B	**0.68**	**0.41**	**0.67**	**0.53**

QI-RAG with Gemma 27B surpasses the best baseline in all metrics, reaching contextual precision of 0.831 (+9.1% over hierarchical), answer relevance of 0.68, and BERTScore of 0.67.

QI-RAG with Gemma 4B surpasses the hierarchical baseline in retrieval precision (0.798 vs. 0.762, +4.7%), but falls below in all generation metrics (Ans. Rel. 0.61 vs. 0.65; METEOR 0.34 vs. 0.37; BERTScore 0.61 vs. 0.64), reflecting the lower synthesis capacity of the 4B model. For local deployment under severe resource constraints, QI-RAG with 4B offers superior retrieval reliability (2.4% failure rate), but final response quality requires at least 12B parameters to surpass the baseline.

Table 4 complements the analysis with generation metrics for all strategies (Gemma 27B + Nomic), showing the full benefit chain: retrieval improvement propagates directly to response quality.

Table 4. Generation quality metrics by strategy (Gemma 27B, Nomic Embed, 500 queries).

Strategy	Ans. Relevance	METEOR	BERTScore	ParaScore
Character-based	0.51	0.22	0.49	0.36
Recursive	0.62	0.34	0.61	0.47
Semantic	0.64	0.36	0.63	0.49
Hierarchical	0.65	0.37	0.64	0.50
QI-RAG	**0.68**	**0.41**	**0.67**	**0.53**

METEOR shows the greatest proportional variation across strategies, indicating closer lexical alignment with reference answers as retrieval quality improves. BERTScore confirms the trend: QI-RAG (0.67) surpasses the character-based baseline (0.49) by 36.7%, indicating semantically closer responses to dataset references.

4.4. Qualitative Analysis

Beyond the aggregate metrics in Sections 4.1–4.3, we manually reviewed QI-RAG's per-query outputs (Gemma 27B + Nomic Embed) to check whether the automated success/failure classification — the same zero-precision criterion behind the Failure Rate metric (Section 4.2) — matches human judgment of the retrieved context and generated response. The 500 queries were split into success/failure pools, and a uniformly random 50% subsample of each was manually inspected. This review is qualitative only, with no additional quantitative tally; it validates that the automated metrics reflect sensible outcomes and surfaces representative cases, one of which follows.

For the query *"What was IBM's gross profit margin in Q3 2022?"*, the **hierarchical baseline** retrieves only the narrative revenue paragraph, missing the margins table, so its response is incomplete. **QI-RAG**, having indexed the question *"What was the gross profit margin reported in Q3 2022?"* from that table's chunk, matches the query at similarity 0.91 and retrieves the header-preserved table, letting the LLM answer precisely (56.3%, matching the dataset reference) — illustrating how question indexing bridges the query-document semantic gap while header preservation keeps retrieved data interpretable.

5. Conclusion

This work investigated the impact of chunking strategies on enterprise financial document retrieval in RAG pipelines, and proposed **Question-Indexed RAG (QI-RAG)**, closing four gaps left open by prior question-based indexing work (Section 2): none of HyDE [Gao et al. 2023], Raina and Gales [Raina and Gales 2024], Eponon [Eponon et al. 2025], or Lee [Lee 2025] were evaluated on enterprise documents with tables and figures, applied type-specific non-textual treatment, used a content-driven adaptive question count over a fixed cap-and-prune policy, or paired the index with a similarity-threshold fallback.

Within the evaluated configurations, chunking strategy is a first-order variable in enterprise RAG performance, outweighing embedding model choice — BGE-M3 and Nomic produced comparable results overall. Baselines show a clear progression: Character (0.536) $<$ Recursive (0.720) $<$ Semantic (0.737) $<$ Hierarchical (0.755) in contextual precision; failure rate drops from 34.4% to 6.1%, confirming that structurally-aware strategies produce more predictable production behavior, consistent with Taiwo and Yusoff's [Taiwo and Yusoff 2026] findings in a different enterprise vertical.

Against this baseline ceiling, **QI-RAG** — the paper's central contribution — achieved contextual precision of 0.831, surpassing the best baseline by 9.1%, and reduced the failure rate to 2.4% (14.3$\times$ below the simplest baseline), directly addressing the fallback and enterprise-domain gaps identified above. Its Overall Accuracy of **0.82** (Section 3.7; not measured for the baselines) further supports this evidence. In generation metrics, QI-RAG with Gemma 27B obtained answer relevance of 0.68 and BERTScore of 0.67. Retrieval gains hold even with Gemma 4B (precision 0.798) — while baselines used Gemma 27B —, indicating the advantage is attributable to the indexing strategy, not just LLM capacity; generation quality gains, however, require at least 12B parameters, a limitation of the generation model, not the indexing approach.

This study has limitations that qualify these findings. Results come from a single

domain (English-language IBM financial reports) and 500 queries, which may not generalize to other enterprise document types or larger corpora. Chunking hyperparameters (800-character size, 200-character overlap) and QI-RAG's retrieval parameters ($\tau = 0.75$, $n = 4$) were held fixed rather than tuned per strategy, so the gaps observed between strategies may shift under different configurations. The fallback mechanism's individual contribution to the failure-rate reduction (Section 3.4.4) was also not isolated via ablation.

Future work includes: (i) evaluating QI-RAG on enterprise corpora from other domains (legal, healthcare, HR); (ii) investigating sensitivity to the number of generated questions per chunk and the fallback threshold, and optimizing the chunking hyperparameters (size and overlap) that were held fixed in this study; and (iii) combining QI-RAG with reranking techniques for further precision optimization in multi-step queries.

References

Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*.

Bennani, S. and Moslonka, C. (2026). A systematic analysis of chunking strategies for reliable question answering. *arXiv preprint arXiv:2601.14123*.

Bruckhaus, T. (2024). RAG does not work for enterprises. *arXiv preprint arXiv:2406.04369*.

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., and Liu, Z. (2024). BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Confident AI (2024). Deepeval: An open-source evaluation framework for LLM systems. Documentation.

Cunha, M. V. d., Silveira, M. R., Santana, B. S., Freitas, L. A., and Corrêa, U. B. (2025). Optimizing and evaluating a retrieval-augmented generation system for normative document retrieval in hospital settings. In *Anais do XXXI Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia 2025)*.

Eponon, A. A., Shahiki-Tash, M., Batyrshin, I., Maldonado-Sifuentes, C. E., Sidorov, G., and Gelbukh, A. (2025). Knowledge compression via question generation: Enhancing multihop document retrieval without fine-tuning. *arXiv preprint arXiv:2506.13778*.

Es, S., James, J., Espinosa-Anke, L., and Schockaert, S. (2024). RAGAS: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2024): System Demonstrations*, pages 150–158.

Gao, L., Ma, X., Lin, J., and Callan, J. (2023). Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Google DeepMind (2025). Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

IBM Research (2024). Docling: An efficient open-source toolkit for AI-ready document conversion. https://github.com/DS4SD/docling.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Lee, S. (2025). Transforming questions and documents for semantically aligned retrieval-augmented generation. *arXiv preprint arXiv:2508.09755*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems (NeurIPS)*.

Nussbaum, Z., Morris, J. X., Duderstadt, B., and Mulyar, A. (2024). Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*.

Qu, R., Tu, R., and Bao, F. (2025). Is semantic chunking worth the computational cost? In *Findings of the Association for Computational Linguistics: NAACL 2025*.

Racine AI (2024). VDR-IBM REAL-MM-RAG: A multimodal retrieval-augmented generation dataset for financial documents. https://huggingface.co/datasets/racineai/VDR_ibm-research_REAL-MM-RAG.

Raina, V. and Gales, M. (2024). Question-based retrieval using atomic units for enterprise RAG. *arXiv preprint arXiv:2405.12363*.

Sarmah, B., Hall, B., Rao, R., Patel, S., Pasquali, S., and Mehta, D. (2024). HybridRAG: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. *arXiv preprint arXiv:2408.04948*.

Sarthi, P., Abdullah, S., Tuli, A., Khanna, S., Goldie, A., and Manning, C. D. (2024). RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*.

Taiwo, S. and Yusoff, M. A. (2026). Evaluating chunking strategies for retrieval-augmented generation in oil and gas enterprise documents. *arXiv preprint arXiv:2603.24556*.

Toresan, B. R., Moreira, V. P., Paula, F. S. F., and Bencke, L. R. (2025). Advanced chunking techniques: A novel approach for semantic splitters. In *Anais do XL Simpósio Brasileiro de Bancos de Dados (SBBD 2025)*.

Wang, X., Wang, Z., Gao, X., Zhang, F., Wu, Y., Xu, Z., Shi, T., Wang, Z., Li, S., Qian, Q., et al. (2024). Searching for best practices in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wang, Z., Gao, C., Xiao, C., Huang, Y., Si, S., Luo, K., Bai, Y., Li, W., Duan, T., Lv, C., Lu, G., Chen, G., Qi, F., and Sun, M. (2025). Document segmentation matters for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. (2023). Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS 2023)*.