Journal of Information and Data Management, 2026, 17:1, doi: 10.5753/jidm.2026.7426
This work is licensed under a Creative Commons Attribution 4.0 International License.

Beyond Trial-and-Error: A Multi-Stage Approach for Selecting Embeddings and Chunking Strategies in RAG

Salvador Ludovico Paranhos [**Universidade Federal de Goiás** | ***salvadorludovico.cs@gmail.com***]
Jonatas Novais Tomazini [**Universidade Federal de Goiás** | ***tomazini@discente.ufg.br***]
Sávio Teles de Oliveira [**Universidade Federal de Goiás** | ***savioteles@inf.ufg.br***]
Celso Camilo Junior [**Universidade Federal de Goiás** | ***celso@inf.ufg.br***]

Informatics Institute, Federal University of Goiás (UFG),
Campus Samambaia - Alameda Palmeiras, s/n - Chácaras Califórnia,
Goiânia - GO, 74690-900, Brazil.

Received: 21 January 2026 • **Published:** 27 November 2026

Abstract The performance of Retrieval-Augmented Generation (RAG) systems critically depends on the interplay between embedding models and text chunking strategies. However, selecting the optimal configuration is often a computationally expensive, trial-and-error process. To address this, we propose a multi-stage optimization framework designed to automate the selection of retriever components. Our approach introduces a preliminary ranking phase that efficiently filters embedding models using lightweight metrics (MRR and Recall) on synthetic queries, significantly reducing the search space before exhaustive indexing. We formally model this selection process as an optimization problem and validate it across three distinct domains: Legal, Audit, and News. Results confirm that the preliminary ranking effectively anticipates downstream performance, preserving top-performing models while discarding suboptimal ones. Furthermore, we extend the evaluation to include stability analysis, revealing a critical trade-off between granularity and reliability: while sentence-level chunking introduces high performance variance (σ) in complex documents, larger window strategies offer superior robustness and predictability. This work provides a generalized, empirically validated pipeline for constructing efficient and stable RAG systems.

Keywords: Retrieval-Augmented Generation (RAG), Retriever Optimization, Embedding Model Selection, Text Chunking Strategies, Stability Analysis

1 Introduction

Retrieval-Augmented Generation (RAG) combines information retrieval with generative models to reduce hallucination and improve factual grounding in knowledge-intensive tasks [Lewis *et al*., 2020]. As these systems are increasingly deployed in high-stakes domains such as legal services, auditing, and public information access, the reliability of the retriever component becomes a critical factor: poor retrieval configurations propagate errors downstream, degrading both answer quality and user trust [Gao *et al*., 2023].

Despite this importance, selecting an appropriate retriever configuration remains a largely manual, trial-and-error process that depends on expert knowledge [Thakur *et al*., 2022]. This creates adoption barriers for organizations lacking specialized teams and risks suboptimal retrieval in domains where configuration choices directly affect system reliability.

The retriever's effectiveness depends primarily on two interrelated components: text segmentation strategies (chunking), which determine how documents are divided into retrievable units, and embedding models, which encode semantic representations of those units [Zhong *et al*., 2025; Liu *et al*., 2025]. The appropriate combination of these components must account for computational constraints, budget limitations, query complexity, and domain-specific language. For instance, sentence-based segmentation may fail in legal texts, where domain-specific abbreviations such as "art." or "sec." mislead standard sentence boundary detection algorithms [Chalkidis *et al*., 2020].

To address this problem, we propose a multi-stage retrieval optimizer that selects the best-performing combination of chunking method and embedding model from a predefined configuration space.

Beyond configuration selection, we conduct a stability analysis across three domains (Legal, Audit, and News), revealing a trade-off between chunking granularity and retrieval reliability: sentence-level strategies introduce high performance variance in complex documents, while larger-window strategies offer more predictable behavior suitable for production deployment.

This paper is organized as follows: Section 2 reviews background and related work; Section 3 details the system architecture and optimization strategy; Section 4 describes the experimental setup; Section 5 presents results and analysis; Section 6 discusses limitations; and Section 7 concludes the paper. [1] The optimizer first discards configurations that are incompatible due to domain or system-level constraints.

[1] This paper is an extended version of "Preliminary Ranking-Based Selection for Optimized Retriever Configuration in RAG Systems", originally published at SBBD 2025 [Paranhos *et al*., 2025]. This version adds a formal optimization model, a deeper stability analysis across domains, and an expanded experimental evaluation.

It then applies a lightweight ranking phase, based on Mean Reciprocal Rank (MRR) and Recall@k computed over synthetic queries, to estimate each embedding model's retrieval capability, significantly reducing the search space before exhaustive indexing. Finally, a grid search over the remaining combinations identifies the configuration that maximizes retrieval quality. To validate the ranking-based filtering, we also evaluate all configurations discarded in the preliminary stage, confirming that the early filter preserves top performers.

2 Background and Related Works

This section presents the foundational components of our approach and discusses related advancements, with a focus on text chunking strategies, embedding models, and recent optimization frameworks for Retrieval-Augmented Generation (RAG) systems.

2.1 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) combines information retrieval with generative models to enhance output accuracy and factual consistency [Lewis *et al*., 2020]. The retriever extracts relevant information from knowledge databases in response to queries, and its effectiveness depends critically on document chunking strategies and embedding quality [Zhong *et al*., 2025]. Retrieval approaches have evolved from token-level retrieval [Khandelwal *et al*., 2020] to entity-based [Nishikawa *et al*., 2022], chunk-based [Ram *et al*., 2023; Reimers and Gurevych, 2019], and graph-based methods [Kang *et al*., 2023].

2.2 Chunking Strategies

Text segmentation (chunking) significantly impacts RAG pipeline performance [Liu *et al*., 2025]. Basic approaches include fixed-size chunking, sliding window with overlap [Safjan, 2023], page-based segmentation for PDFs, and hierarchical methods using structural delimiters. Advanced strategies include Parent Document Retrieval [Antematter team, 2024], Metadata Filtering, and Context-Enriched Chunking.

2.3 Embedding Models

Dense vector retrieval maps texts into a continuous embedding space, positioning semantically similar documents and queries closer together [Lewis *et al*., 2020]. Current models such as `E5` and `NV-Embed-v2` achieve strong results on MTEB benchmarks [Muennighoff *et al*., 2022], while `KEPLER` enhances embeddings with knowledge graphs [Wang *et al*., 2021]. Although alternatives exist (BM25, ColBERT), our work focuses on dense embeddings.

2.4 RAG Pipeline Optimization and Evaluation

Recent work has addressed the challenge of systematically optimizing and evaluating RAG pipelines. AutoRAG [Kim *et al*., 2024] proposes an AutoML-style framework that exhaustively explores combinations of RAG modules, including query expansion, retrieval, reranking, and generation, to identify the best pipeline for a given dataset. AutoRAG-HP [Fu *et al*., 2024] formulates hyperparameter tuning (e.g., top-k documents, prompt compression ratio, embedding method) as an online multi-armed bandit problem, achieving competitive performance with significantly fewer evaluations than grid search. On the evaluation side, ARES [Saad-Falcon *et al*., 2024] introduces an automated framework that fine-tunes lightweight language model judges on synthetic data to assess context relevance, answer faithfulness, and answer relevance, reducing the need for extensive human annotation. FlashRAG [Jin *et al*., 2025] provides a modular open-source toolkit with standardized benchmarks and pre-implemented algorithms for reproducible RAG research. CRUD-RAG [Lyu *et al*., 2024] extends the evaluation scope beyond question-answering by categorizing RAG applications into Create, Read, Update, and Delete scenarios, evaluating the impact of retrieval components and knowledge base construction across diverse tasks.

Unlike these prior works, which either optimize full pipelines through exhaustive or bandit-based search [Kim *et al*., 2024; Fu *et al*., 2024], focus on evaluation frameworks [Saad-Falcon *et al*., 2024; Lyu *et al*., 2024], or provide general-purpose toolkits [Jin *et al*., 2025], our approach specifically targets the retriever configuration problem: the joint selection of embedding model and chunking strategy. We introduce a lightweight preliminary ranking phase based on classical IR metrics (MRR and Recall@k) that efficiently filters embedding models before exhaustive evaluation, significantly reducing the search space. Furthermore, we provide an empirical stability analysis across domains, revealing the trade-off between chunking granularity and retrieval reliability, an aspect that, to the best of our knowledge, has not been systematically studied in prior RAG optimization literature.

3 System Architecture and Optimization Strategy Based on Preliminary Ranking

The primary objective of this work is to optimize document retrieval in Retrieval-Augmented Generation (RAG) systems by selecting the best combination of text segmentation strategy and embedding model. To this end, we propose a three-phase optimization architecture: (i) preliminary ranking of embedding models based on ranking metrics, (ii) document indexing using multiple text segmentation strategies and the top-n embedding models from phase (i), and (iii) exhaustive evaluation of the remaining combinations. The architecture code and datasets are publicly available [2].

[2] `https://github.com/salvadorludovico/retriever-optmizer`

3.1 Problem Formalization

We model the retriever configuration selection as an optimization problem over a discrete search space [Gao *et al*., 2023]. Let $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ be the set of documents within a specific domain, and let $\mathcal{Q} = \{q_1, q_2, \ldots, q_m\}$ be a set of queries derived from $\mathcal{D}$, used to evaluate retrieval performance. We define the configuration space Ω as the Cartesian product of the set of segmentation methods $\mathcal{C}$ and the set of embedding models $\mathcal{E}$:

$$\Omega = \mathcal{C} \times \mathcal{E} \tag{1}$$

A configuration $k \in \Omega$ is a pair (c_i, e_j), where $c_i \in \mathcal{C}$ represents a chunking strategy and $e_j \in \mathcal{E}$ represents an embedding model. The preliminary ranking phase acts as a filtering function $\Phi(\mathcal{E}, \mathcal{D}, \mathcal{Q})$, reducing the search space to a subset $\Omega' \subset \Omega$ by retaining only the embedding models that satisfy a performance criterion (e.g., MRR $> \theta$ or top-N rank). The relative performance of embedding models can be compared using standardized benchmarks [Muennighoff *et al*., 2022].

The evaluation function $\mathit{Score}(k, \mathcal{D}, \mathcal{Q})$ for a given configuration k is defined as a weighted combination of four RAG evaluation metrics computed over $\mathcal{Q}$:

$$\begin{aligned}\mathit{Score}(k, \mathcal{D}, \mathcal{Q}) = \alpha \cdot R(k, \mathcal{Q}) + \beta \cdot F(k, \mathcal{Q}) \\ + \gamma_1 \cdot P_{\text{ref}}(k, \mathcal{Q}) + \gamma_2 \cdot P_{\text{noref}}(k, \mathcal{Q})\end{aligned} \tag{2}$$

where $R(k, \mathcal{Q})$ denotes Context Recall, $F(k, \mathcal{Q})$ denotes Faithfulness, $P_{\text{ref}}(k, \mathcal{Q})$ denotes reference-based Context Precision, and $P_{\text{noref}}(k, \mathcal{Q})$ denotes reference-free Context Precision. The coefficients α, β, γ_1, and γ_2 are domain-specific hyperparameters subject to the constraint $\alpha + \beta + \gamma_1 + \gamma_2 = 1$, reflecting the relative importance of each metric. Their concrete values are defined in Section 4.

The objective of the optimizer is to find the configuration k^* that maximizes the score within the reduced search space:

$$k^* = \underset{k \in \Omega'}{\operatorname{argmax}}\, \mathit{Score}(k, \mathcal{D}, \mathcal{Q}) \tag{3}$$

This formulation follows the general framework for RAG configuration search described by Gao *et al*. [2023].

3.2 Queries Generation

Synthetic (query, reference document) pairs were generated using `Gemini 2.0 Flash` applied to document excerpts from each domain corpus. The model was prompted to emulate realistic information needs from domain experts — legal professionals for STJ, auditors and fiscal analysts for TCM-GO, and general readers for Folha de S. Paulo. For each document, the model was instructed to produce: (i) a technically precise query reflecting a plausible real-world information need; (ii) a concise reference answer; and (iii) the supporting passage from the source document. Prompts were tailored per domain to enforce specificity and discourage generic or vague queries.

To ensure thematic coverage without redundancy, documents were grouped by topic and representative documents were selected from each group, over which queries were generated. The generated queries were subsequently reviewed manually to verify quality and domain alignment, with low-quality or repetitive queries discarded. A small number of real-world queries provided by domain experts supplemented the synthetic data. All query-reference pairs are publicly available in the project repository[2].

Table 1 presents one representative example per domain, illustrating the specificity and technical depth of the generated queries.

Table 1. Representative query examples per domain.

Domain	Query Example
STJ	*"O STJ admite a aplicação do princípio da insignificância nos crimes de furto simples em réus reincidentes?"*
TCM-GO	*"A CGM de Goianésia apresentou insuficiência na capacitação de servidores voltada para auditoria interna governamental no período de 2019 a 2022?"*
Folha	*"Quais foram os principais argumentos utilizados pelo governo federal para justificar a reforma tributária aprovada em 2023?"*

Algorithm 1: Preliminary Ranking-Based Evaluation for Embedding Selection

Input : TOP_K, embedding model, data = [($query_1$, ref_1), ...]
Output : MRR, Recall

```
corpus ← RemoveDuplicates(data)
model ← LoadModel(embedding_model)
corpusEmbeddings ← EncodeCorpus(model, corpus)
results ← empty list
foreach (query, reference) ∈ data do
    queryEmb ← EncodeQuery(model, query)
    similarities ← CosineSimilarity(queryEmb, corpusEmbeddings)
    ranked ← SortDescending(similarities)
    refPosition ← FindPosition(ranked, reference)
    rank ← refPosition + 1
    Add (query, ref, rank, rank ≤ TOP_K, 1/rank) to results
MRR ← AverageReciprocalRank(results)
Recall ← AverageRecallAtK(results)
```

3.3 Preliminary Model Selection Phase

Prior to exhaustive evaluation, an initial filtering stage estimates each embedding model's ranking capability using the synthetic (query, reference) pairs described in Section 3.2. Two metrics are employed: Mean Reciprocal Rank (MRR) and Recall@k. For each pair, cosine similarity is computed between the query embedding and all document embeddings; documents are ranked by descending similarity, and the position of the reference document determines the metric scores. The complete procedure is formalized in Algorithm 1.

This phase enables the selection of the most promising models based on their retrieval quality, thereby avoiding exhaustive evaluation of all candidates during the final grid

search.

3.4 Indexing Phase with Segmentation Strategies

Following the preliminary ranking of embedding models, the indexing phase begins. Each domain-specific dataset is segmented using multiple chunking methods, resulting in smaller text units. These chunks are then encoded with the embedding models selected in the previous phase, producing dense vector representations. The combination of a chunking method and an embedding model, referred to as a *configuration*, results in a unique collection of vectors, which is independently indexed in the vector store.

This design ensures that all relevant combinations of chunking and embedding are made available for retrieval during the evaluation phase, while restricting the scope exclusively to the embedding models identified as most promising in the previous step. The overall workflow of this phase is depicted in Figure 1.

3.5 Evaluation and Recommendation Phase

The final stage of the architecture applies an exhaustive search (grid search) over the selected combinations of segmentation methods and embedding models derived from the preliminary ranking phase. The objective is to identify, within this reduced and promising subset, the best system configuration for each domain.

In this phase, synthetic queries are transformed into vectors using the same models employed during indexing. Each query vector is compared with the corresponding vector collections, and the five most similar documents are retrieved based on cosine similarity. The retrieved documents are then compared with the expected reference documents.

Retrieval operates at the chunk level: given a query, the vector store returns the top-5 most similar chunks across the entire indexed collection, ranked by cosine similarity. Document-level ranking is implicitly determined by the highest-scoring chunk of each document, that is, a document is considered retrieved if any of its chunks appears among the top-5 results (max-pooling aggregation). In cases where multiple chunks from the same document are retrieved, the document's rank is defined by its best-scoring chunk. This strategy was adopted because RAG pipelines consume individual chunks as context for generation, not full documents, making chunk-level relevance the operationally meaningful unit of evaluation. This approach introduces a known bias: documents with more chunks have a higher prior probability of appearing in top-k results. However, since this effect is consistent across all evaluated configurations, it does not affect the relative comparison between models or chunking strategies.

At the end of the evaluation, as defined in Eq. 2, the decision function is applied to determine the most efficient configuration, that is, the combination of chunking method and embedding model that achieves the highest average performance within each domain. Configurations are grouped by arithmetic means of the obtained metrics, which are subsequently combined into a single value through a weighted average to identify the optimal parameter set k^*.

4 Experimental Materials and Methods

This section presents the datasets, experimental setup, and evaluation metrics used to validate the retrieval optimization strategies described in the previous section.

4.1 Datasets

We used three datasets from different domains, indexed in a vector store for retrieval tasks:

- **STJ Jurisprudence** — 8,000 documents containing decisions and interpretations from Brazil's Superior Court of Justice.
- **TCM-GO Audit Reports** — 38 financial audit reports and rulings issued by Goiás' Municipal Court of Accounts.
- **Folha de S. Paulo Articles** — 8,000 news articles covering politics, economy, science, and society.

The number of documents and queries per domain reflects domain-specific characteristics. For STJ and Folha, the full corpora contain approximately 60,000 documents each; we sampled 8,000 documents, which we considered sufficient to represent the semantic diversity of the full corpus. Queries were not generated uniformly per document — instead, documents were grouped thematically and queries were generated over representative documents from each group, avoiding redundancy while preserving semantic coverage. The variation in query counts reflects structural differences between domains: STJ documents are short and thematically homogeneous, requiring fewer representatives to cover the corpus adequately (146 queries); Folha spans broader topics requiring more representatives (340 queries); and TCM-GO, despite only 38 documents, consists of dense audit reports with multiple findings per document, each contributing several queries (152 queries total). Each query was linked to a reference document to enable supervised evaluation. The generation methodology and concrete examples are detailed in Section 3.

4.2 Experimental Setup

The experiments were implemented in Python using `LangChain` and Hugging Face Transformers for embedding generation. All models were run on a NVIDIA DGX station with two CUDA-enabled V100 GPUs (32 GB VRAM), allowing dynamic resource allocation and efficient parallel processing. Loading and preprocessing involved text extraction with `pdfplumber`, normalization, and structuring via `pandas`; `Pgvector` PostgreSQL extension was used for vector storage and search engine. Document chunking was applied prior to vectorization, considering token limits and semantic coherence, supported by `LangChain` tools.

Six sentence-transformer-based embedding models were initially considered, from which three were

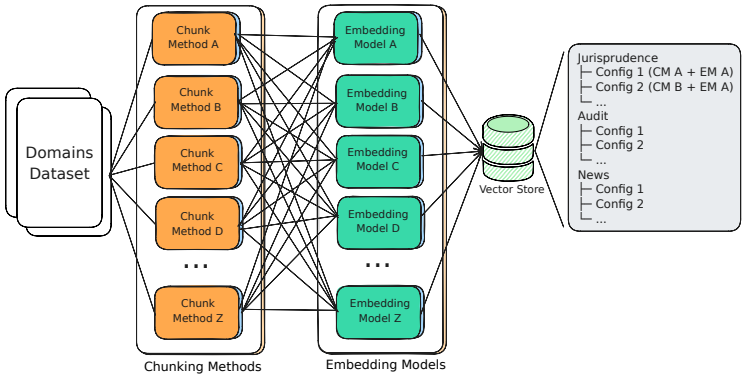

Figure 1. Indexing phase: document chunking and embedding generation.

selected based on multilingual support, instruction tuning, and recent benchmark performance: `BGE-M3`, `multilingual-e5-large`, and `KaLM`.[3] These three models were combined with multiple segmentation strategies across three domains. For case law (STJ) and news articles (*Folha de S. Paulo*), five chunking methods were used: `hierarchical-sentence`, `sliding-window-1024`, `chunk-fixed-size-512`, `hierarchical-section`, and `hierarchical-paragraph`, resulting in 15 configurations per domain. For audit reports (TCM-GO), the same embeddings were tested including an additional method, `per-page-chunk`, resulting in a total of 18 combinations.

The `per-page-chunk` strategy was included exclusively for the audit domain because TCM-GO documents are long institutional PDF reports (often exceeding 100 pages) with high informational density and strong contextual interdependence between audit findings, legal criteria, and compliance evidence. In such documents, fine-grained chunking strategies frequently fragment these interconnected elements, breaking the argumentative continuity required for meaningful retrieval. Page-level segmentation acts as a domain-aware, large-window strategy that preserves broader context within each chunk. This method was not applied to the STJ and *Folha de S. Paulo* corpora, as those documents follow a continuous narrative structure where page boundaries do not correspond to meaningful semantic divisions.

4.3 Evaluation metrics

To aggregate the metrics, we applied a weighted average (Eq. 2), with weights defined through a principled qualitative analysis conducted by the authors, grounded in the practical role and relative importance of each metric in RAG-based systems. The hyperparameters were set as follows: $\alpha = 0.35$ (Context Recall), $\beta = 0.30$ (Faithfulness), $\gamma_1 = 0.25$ (reference-based Context Precision), and $\gamma_2 = 0.10$ (reference-free Context Precision). Context Recall received the highest weight due to its strong influence on answer completeness, followed by Faithfulness, which ensures answer reliability. The reference-based variant of Context Precision received a higher weight than the no-reference variant, as it is considered more stable. These choices reflect an informed understanding of the trade-offs involved in retrieval-based evaluation and aim to balance completeness, precision, and factual grounding [Es *et al.*, 2023].

Throughout the results tables, we abbreviate these metrics as WA (Weighted Average), F (Faithfulness), CR (Context Recall), and CP (Context Precision).

[3] Full Hugging Face identifiers: `BAAI/bge-m3`, `intfloat/multilingual-e5-large`, and `HIT-TMG/KaLM-embedding-multilingual-mini-instruct-v1.5`.

5 Results and Discussion

This section analyzes the results obtained through the proposed optimization pipeline. The discussion is structured by domain and evaluates whether the preliminary ranking metrics (MRR and Recall@5) effectively anticipated the best-performing configurations in a full grid search. Table 2 summarizes the preliminary ranking values for the three top-ranked models across all domains and serves as a consolidated reference throughout this section.

5.1 News Domain Performance

The complete results for the news domain are shown in Table 3. The best configuration was `BGE-M3` + `sliding-window-1024` (WA = 86.77%, CR = 92.51%). All three models performed well with large-window strategies, but `KaLM` showed higher sensitivity to granularity changes, with WA dropping below 60% under `hierarchical-sentence`. The preliminary ranking order (Table 2) was fully preserved: `BGE-M3` led both phases (MRR = 88.74; Recall@5 = 94.12), confirming that the early-stage filtering reliably anticipates downstream performance in this domain.

Table 2. Performance of embedding models by domain

Domain	Model	MRR (%)	Recall@5 (%)
Jurisprudence	`BGE-M3`	**91.51**	**97.95**
	`multilingual-e5-large`	89.37	95.21
	`KaLM`	86.18	94.52
Audit	`BGE-M3`	**50.67**	**63.82**
	`KaLM`	45.99	61.84
	`multilingual-e5-large`	47.48	60.53
News	`BGE-M3`	**88.74**	**94.12**
	`KaLM`	77.70	92.94
	`multilingual-e5-large`	75.89	85.88

Table 3. Results per configuration for the *Folha de S. Paulo* dataset. Bold values denote the highest score per column.

Embedding Model	Chunk Method	WA (%)	F (%)	CR (%)	CP (Ref) (%)	CP (No Ref) (%)
BGE-M3	sliding-window-1024	**86.77**	**86.03**	**92.51**	**81.64**	**81.73**
multilingual-e5-large	sliding-window-1024	84.39	85.27	89.89	78.04	78.36
KaLM	sliding-window-1024	83.84	83.26	89.39	78.67	79.11
BGE-M3	hierarchical-section	83.58	80.14	91.62	78.56	78.30
multilingual-e5-large	hierarchical-section	75.72	73.17	81.33	72.32	72.25
BGE-M3	chunk-fixed-size-512	68.89	61.93	74.53	69.25	69.12
BGE-M3	hierarchical-paragraph	66.12	58.27	70.74	68.12	68.53
KaLM	hierarchical-section	65.49	61.87	71.61	62.04	63.54
KaLM	chunk-fixed-size-512	59.34	52.55	64.27	60.50	59.52
KaLM	hierarchical-paragraph	57.84	49.80	62.30	60.37	60.04
multilingual-e5-large	chunk-fixed-size-512	56.91	50.80	62.95	55.95	56.53
multilingual-e5-large	hierarchical-paragraph	56.78	50.09	57.83	61.06	62.51
BGE-M3	hierarchical-sentence	48.84	36.99	47.23	60.93	59.82
multilingual-e5-large	hierarchical-sentence	44.50	35.01	42.63	54.30	54.98
KaLM	hierarchical-sentence	41.34	29.98	39.79	52.47	52.99

5.2 Legal Domain Performance

The legal domain comprises documents characterized by highly formal structure, high informational density, and strong semantic continuity. Full results are presented in Table 4. The best configuration was BGE-M3 + chunk-fixed-size-512 (WA = 91.49%, CR = 96.99%). Notably, BGE-M3 proved robust across all granularities, with three different chunking strategies exceeding WA = 90%. The other models were competitive in structured configurations (e.g., multilingual-e5-large reached WA = 90.02% with hierarchical-paragraph) but exhibited sharper performance declines under less cohesive methods. The preliminary ranking order (Table 2) was again fully preserved, with BGE-M3 leading (MRR = 91.51; Recall@5 = 97.95), followed by multilingual-e5-large and KaLM.

5.3 Audit Domain Performance

Grid search results (Table 5) confirm BGE-M3 + sliding-window-1024 as the best configuration (WA = 84.64%, CR = 92.3%). KaLM and multilingual-e5-large underperformed (best WA = 81.32% and 79.1%, respectively), while all models showed substantial drops with hierarchical-sentence, confirming that fine-grained chunking impairs retrieval in dense audit documents.

The preliminary ranking order (Table 2) was preserved: BGE-M3 led both phases (MRR = 50.67; Recall@5 = 63.82). To assess the generalizability of the preliminary ranking strategy beyond the three core models, we conducted an additional experiment with three embedding models not included in the main evaluation: nomic-embed-text-v1, mxbai-embed-large-v1, and flan-t5-large. These models were selected from the MTEB leaderboard to cover diverse architectural families: a bidirectional encoder (nomic-embed-text-v1), a large-scale embedding model (mxbai-embed-large-v1), and an encoder-decoder architecture (flan-t5-large), thereby testing whether the ranking mechanism remains effective across different model designs. Results (Table 6) confirmed the strategy's applicability: nomic-embed-text-v1 ranked first (MRR = 39.92; Recall@5 = 52.63%), followed by mxbai-embed-large-v1 (31.22; 37.50%) and flan-t5-large (21.24; 28.95%).

This segmentation was previously identified as effective in the audit domain (see Table 5), enabling consistent estimation of each model's WA. The resulting values were 76.7% for nomic-embed-text-v1, 69.3% for mxbai-embed-large-v1, and 53.6% for flan-t5-large. This strategy enables efficient screening of promising models without the need for exhaustive evaluation, maintaining strong alignment with performance estimated using a reference configuration.

5.4 Stability and Granularity Impact Analysis

Average performance metrics alone do not capture retriever reliability. Standard deviation (σ) and min–max range reveal consistency trade-offs that are critical for production deployment Buckley and Voorhees [2000]; Sakai [2006]. While BGE-M3 consistently leads in average scores, stability analysis shows that the choice of chunking strategy significantly affects output predictability.

In the Jurisprudence domain, the configuration BGE-M3 + chunk-fixed-size-512 achieves the highest average (0.9292) with the lowest observed variance ($\sigma = 0.1104$). The hierarchical-paragraph strategy reaches a comparable average (0.9243) but with higher dispersion ($\sigma = 0.1368$). This indicates that fixed-size chunking, despite being semantically naïve, provides more consistent segmentation boundaries in legal texts, where paragraph lengths vary widely, reducing the incidence of retrieval failures below a minimum quality threshold.

Across all three domains, sentence-level strategies consistently present the highest standard deviations, confirming a negative correlation between chunk granularity and performance variance. As detailed in Table 7 and Figure 2, in the Audit domain BGE-M3 + hierarchical-sentence exhibits $\sigma = 0.2915$, compared to $\sigma = 0.2072$ for the sliding-window-1024 approach, an approximate 40% increase in variance. Sentence-level chunks frequently lack sufficient context to resolve the dense, interdependent terminology of audit reports, resulting in a bimodal score distri-

Table 4. Results per configuration for the *STJ Jurisprudence* dataset. Bold values denote the highest score per column.

Embedding Model	Chunk Method	WA (%)	F (%)	CR (%)	CP (Ref) (%)	CP (No Ref) (%)
BGE-M3	chunk-fixed-size-512	**91.49**	83.98	**96.99**	92.51	92.25
BGE-M3	hierarchical-paragraph	90.99	**86.21**	92.94	**93.04**	**93.39**
BGE-M3	sliding-window-1024	90.03	86.12	94.79	88.63	88.56
multilingual-e5-large	hierarchical-paragraph	90.02	84.15	94.56	90.46	90.59
KaLM	hierarchical-paragraph	87.27	79.09	92.48	89.19	88.80
multilingual-e5-large	sliding-window-1024	87.01	82.99	94.91	82.26	83.29
BGE-M3	hierarchical-sentence	86.75	77.66	88.31	92.93	93.14
multilingual-e5-large	hierarchical-sentence	86.16	76.54	89.81	90.50	91.38
multilingual-e5-large	chunk-fixed-size-512	85.98	77.78	91.78	87.05	87.61
KaLM	sliding-window-1024	85.82	83.8	93.52	79.89	79.77
BGE-M3	hierarchical-section	85.36	79.82	92.48	82.98	83.07
multilingual-e5-large	hierarchical-section	82.05	77.88	91.32	76.32	76.41
KaLM	hierarchical-sentence	81.48	70.57	84.61	87.83	87.36
KaLM	chunk-fixed-size-512	80.62	72.30	86.92	81.45	81.43
KaLM	hierarchical-section	60.83	54.16	70.14	57.20	57.34

Table 5. Results per configuration for the *TCM-GO Audit Reports* dataset. Bold values denote the highest score per column.

Embedding Model	Chunk Method	WA (%)	F (%)	CR (%)	CP (Ref) (%)	CP (No Ref) (%)
BGE-M3	sliding-window-1024	**84.64**	**79.42**	**92.3**	81.47	81.37
BGE-M3	per-page-chunk	83.19	75.53	90.11	**82.66**	**83.33**
BGE-M3	hierarchical-section	81.53	74.03	87.09	82.33	82.58
KaLM	per-page-chunk	81.32	72.66	89.87	79.83	81.13
KaLM	sliding-window-1024	81.29	73.25	88.68	81.08	80.09
multilingual-e5-large	sliding-window-1024	79.10	70.9	90.57	74.71	74.50
KaLM	hierarchical-section	74.96	65.06	82.46	75.84	76.25
BGE-M3	chunk-fixed-size-512	72.33	57.09	81.03	76.78	76.50
multilingual-e5-large	per-page-chunk	68.81	58.95	81.2	64.64	65.43
BGE-M3	hierarchical-sentence	67.56	52.67	75.72	72.23	71.94
BGE-M3	hierarchical-paragraph	67.40	48.26	76.90	74.38	74.13
multilingual-e5-large	chunk-fixed-size-512	67.14	51.32	78.38	69.22	70.05
KaLM	chunk-fixed-size-512	66.08	46.36	76.15	73.17	72.27
KaLM	hierarchical-paragraph	65.04	46.01	77.52	68.56	69.63
multilingual-e5-large	hierarchical-section	64.90	55.27	77.25	60.64	61.25
multilingual-e5-large	hierarchical-paragraph	62.66	42.38	75.67	67.22	66.51
KaLM	hierarchical-sentence	57.46	35.90	67.93	65.87	64.52
multilingual-e5-large	hierarchical-sentence	55.31	35.67	61.98	65.52	65.36

Table 6. Performance of embedding models using MRR, Recall@5, and weighted average (without optimizer).

Model	MRR (%)	Recall@5 (%)	Weighted Avg. (%)
nomic-embed-text-v1	**39.92**	**52.63**	76.7
mxbai-embed-large-v1	31.22	37.50	69.3
flan-t5-large	21.24	28.95	53.6

bution: the retriever either matches the relevant passage precisely or fails to retrieve it at all. For compliance-oriented applications where consistent recall is a requirement, this volatility is a practical disqualifier.

Furthermore, the Domain Sensitivity analysis reveals that document complexity acts as an instability multiplier. The News domain, characterized by diverse topics and less rigid structure, is inherently the most volatile, with even the best configuration (`sliding-window-1024`) showing an elevated standard deviation ($\sigma = 0.2466$) compared to the best Legal configuration ($\sigma = 0.1104$). However, the impact of inadequate chunking is most severe in the Audit domain. When shifting from large windows to sentence-level chunks in audit reports, the standard deviation increases from 0.2072 to 0.2915, whereas in the Legal domain the shift is less dramatic (from 0.1104 to 0.1561). This empirical evidence suggests that in domains with a high degree of "informational entanglement," such as auditing—where a finding strongly depends on surrounding context—the choice of chunking strategy is not merely an accuracy optimization, but a fundamental requirement for system stability.

To verify whether the observed distributional differences reflect statistically significant distinctions rather than sampling variability, we applied the Wilcoxon signed-rank test — a non-parametric paired test that does not assume normality — to per-query score distributions for the most contrasting configurations across all three domains. In the Audit domain, the comparison between BGE-M3 + sliding-window-1024 and BGE-M3 + hierarchical-sentence yielded $W = 1758.0$, $p < 0.0001$, confirming that the performance gap (mean 0.845 vs. 0.712) is statistically significant. This result holds consistently across all models (KaLM: $p < 0.0001$; multilingual-E5: $p < 0.0001$). Notably, the comparison between the two top-performing configurations — sliding-

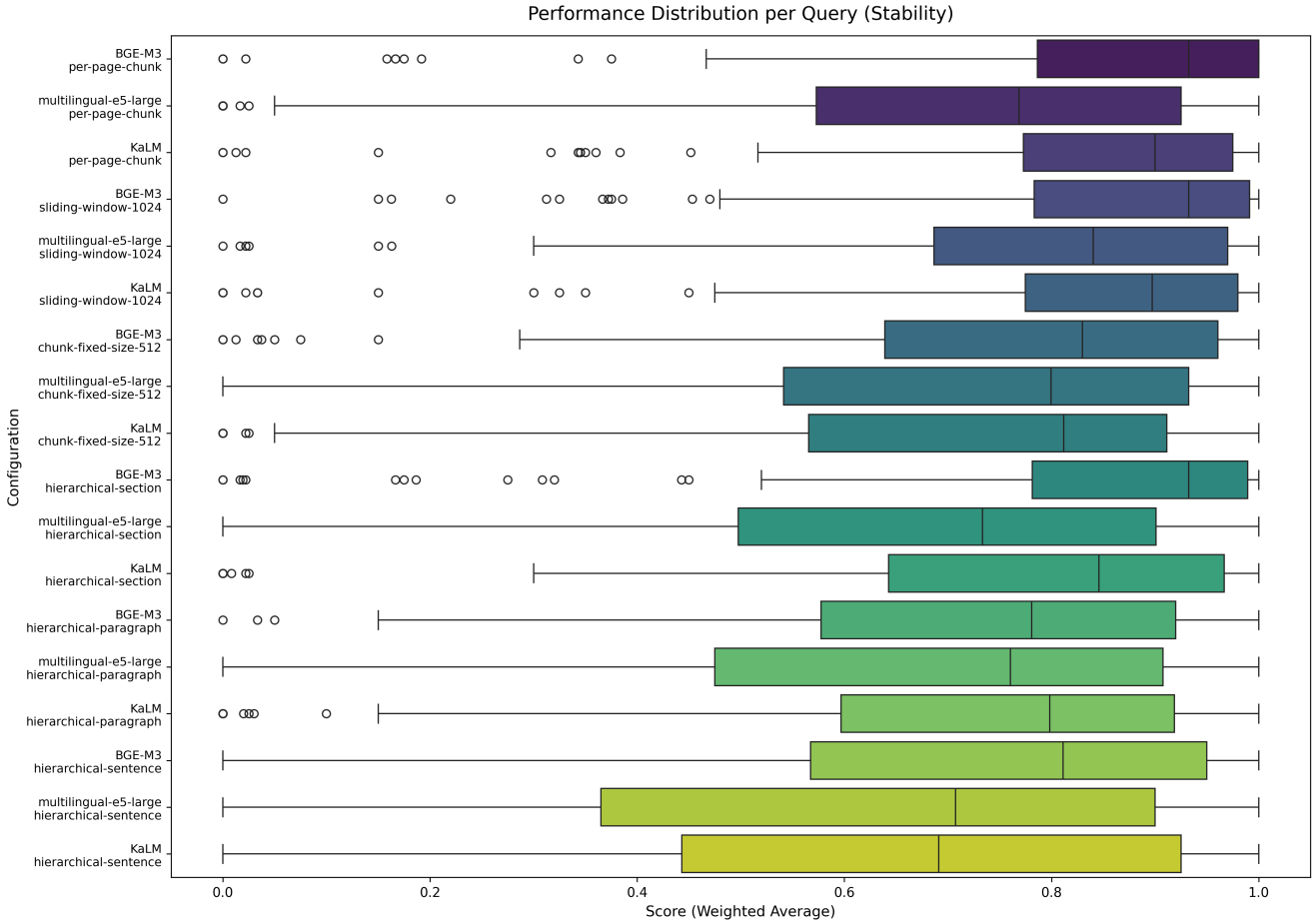

Figure 2. Performance distribution per query in the Audit domain. The boxplots illustrate the stability trade-off: while sliding-window approaches (top) maintain compact interquartile ranges, sentence-level strategies (bottom) exhibit high dispersion and frequent lower-bound outliers.

window-1024 and per-page-chunk — yielded $p = 0.973$, indicating that these strategies are statistically equivalent and that either can be recommended for production use in the audit domain.

In the News domain, all tested pairs involving hierarchical-sentence yielded $p < 0.0001$ regardless of model, confirming the severity of fine-grained chunking in this domain. In the Jurisprudence domain, the top vs. bottom comparison (chunk-fixed-size-512 vs. hierarchical-sentence) did not reach significance ($p = 0.123$), reflecting the structural regularity of legal documents, which preserves retrieval quality even under finer segmentation — a finding that is itself informative about domain-specific robustness. Taken together, these results statistically validate the stability analysis presented in this section and the practical recommendations derived from it.

5.5 Analysis of Embedding Model Architecture: The Case of `BGE-M3`

The consistent superiority of `BGE-M3` across all domains can be attributed to its hybrid architecture, which supports dense, sparse, and multi-vector retrieval within a unified framework Liu *et al*. [2024]. By producing multiple representations per input via multi-CLS pooling and supporting sequences up to 8,192 tokens, the model jointly captures semantic similarity, lexical matching, and contextual signals. This design is particularly beneficial for large-span chunking strategies in dense informational domains. Additionally, self-knowledge distillation across retrieval heads and contrastive training on large-scale multi-domain corpora improve cross-domain generalization Liu *et al*. [2024], which is consistent with MTEB benchmark results where `BGE-M3` outperforms `multilingual-e5-large` and `KaLM` Muennighoff *et al*. [2023].

6 Limitations of the preliminary ranking Strategy

Although the results confirm the effectiveness of the preliminary ranking strategy in reducing the search space and preserving high-performing configurations, the approach depends directly on the quality and representativeness of the synthetic query-reference set used in the initial evaluation phase. Since embedding model selection is guided by ranking metrics computed from this set, any thematic bias or limited semantic coverage may compromise the accuracy of the pre-selection. Moreover, because synthetic queries are derived from the same document corpus, there is a risk of local overfitting; a model may rank extracted pairs effectively but fail to generalize to real or future system inputs. Mitigating this limitation requires careful construction of the query-reference dataset, ensuring sufficient diversity and domain

Table 7. Stability Analysis for Audit Domain: Comparison of Top and Bottom Configurations (sorted by Mean Score).

Model	Configuration	Mean	Std (σ)
High Stability (Top Performers)			
BGE-M3	sliding-window-1024	**0.8449**	**0.2072**
BGE-M3	per-page-chunk	0.8438	0.2181
KaLM	per-page-chunk	0.8252	0.2201
Low Stability (Sentence/Granular)			
BGE-M3	hierarchical-sentence	0.7124	0.2915
KaLM	hierarchical-sentence	0.6308	0.3212
multilingual-e5-large	hierarchical-sentence	0.6142	0.3183

alignment. Enrichment with real usage examples and iterative evaluation of semantic coverage are promising directions to enhance the robustness of the method.

7 Conclusion

This work proposed a modular framework for optimizing retrievers in Retrieval-Augmented Generation (RAG) systems, based on the preliminary ranking of embedding models using classical ranking metrics. The central objective was to reduce the search space during exhaustive configuration exploration while preserving the quality of the resulting solutions. The architecture was structured into three stages — preliminary ranking, segmented indexing, and exhaustive evaluation — enabling the application of progressive filters to efficiently identify the best combination of embedding model and text segmentation strategy.

The pre-selection stage, guided by metrics such as Mean Reciprocal Rank (MRR) and Recall@k, proved to be an effective tool for anticipating downstream retrieval performance. Experiments conducted across three distinct domains — audit, news, and legal — empirically validated this approach, showing that the top-ranked models in preliminary evaluations maintained superior performance in subsequent stages, even under varying granularities and heterogeneous textual structures. The main contributions of this study are: (i) the formalization of an optimization pipeline for retrievers based on modular, empirically-driven stages; (ii) the introduction of a lightweight preliminary ranking mechanism that significantly reduces the search space; and (iii) the experimental validation of the approach in diverse scenarios, demonstrating its generalizability and practical applicability.

Additionally, stability analysis revealed that larger chunking strategies provide significantly lower variance in complex domains, ensuring greater reliability for production environments.

As future work, we propose to explore extensions of the optimizer through hybrid retrieval approaches (e.g., sparse or supervised models) and, in later stages, develop complementary mechanisms for answer generation, integrating a Generation Optimizer into the full RAG system pipeline.

Acknowledgements

We acknowledge the Center of Excellence in Artificial Intelligence (CEIA) for providing the research environment and infrastructural support necessary for the development of this work.

Funding

This study was financed by the Court of Accounts of State of Goiás Municipalities (TCM-GO) and the Court of Accounts of the State of Goiás (TCE-GO). This work was also supported by the National Institute of Science and Technology (INCT) in Responsible Artificial Intelligence for Computational Linguistics and Information Treatment and Dissemination (TILD-IAR) under grant number 408490/2024-1.

Authors' Contributions

SL and JN contributed equally to the conceptualization, methodology, project administration, data curation, investigation, data visualization, and writing (original draft, review, and editing) of the manuscript. CC and ST contributed to conceptualization, supervision, resource allocation, and manuscript review and editing.

Competing interests

The authors declare no conflict of interest.

Availability of data and materials

The code built and datasets generated and/or analysed during the current study are available in the GitHub repository, `https://github.com/salvadorludovico/retriever-optmizer`.

References

Antematter team (2024). Optimizing retrieval-augmented generation with advanced chunking techniques: A comparative study. Accessed: 2025-03-31.

Buckley, C. and Voorhees, E. M. (2000). Evaluating evaluation measure stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40. ACM.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. (2020). LEGAL-BERT: The muppets straight out of law school. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904,

Online. Association for Computational Linguistics. DOI: 10.18653/v1/2020.findings-emnlp.261.

Es, S., James, J., Espinosa-Anke, L., and Schockaert, S. (2023). Ragas: Automated evaluation of retrieval augmented generation.

Fu, J., Qin, X., Yang, F., Wang, L., Zhang, J., Lin, Q., Chen, Y., Zhang, D., Rajmohan, S., and Zhang, Q. (2024). Autorag-hp: Automatic online hyper-parameter tuning for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3875–3891. DOI: 10.18653/v1/2024.findings-emnlp.223.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Jin, J., Zhu, Y., Dou, Z., and Wen, J.-R. (2025). Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM Web Conference 2025 (WWW)*. DOI: 10.1145/3701716.3715313.

Kang, J., Lee, J., and Kang, J. (2023). Knowledge graph-augmented language models for knowledge-grounded dialogue generation. *arXiv preprint arXiv:2305.18846*.

Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., and Lewis, M. (2020). Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Kim, D., Kim, B., Han, D., and Eibich, M. (2024). Autorag: Automated framework for optimization of retrieval augmented generation pipeline. *arXiv preprint arXiv:2410.20878*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Fan, A., Chaudhary, V., Rocktäschel, T., and Riedel, S. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*.

Liu, X., Zhou, Z., Zhou, T., Sun, M., and Wang, T. (2024). Bge-m3: A multi-function embedding model for dense, sparse and multi-vector retrieval. *arXiv preprint arXiv:2402.03216*.

Liu, Z., Simon, C.-E., and Caspani, F. (2025). Passage segmentation of documents for extractive question answering.

Lyu, Y., Li, Z., Niu, S., Xiong, F., Tang, B., Wang, W., Wu, H., Liu, H., Xu, T., and Chen, E. (2024). Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *ACM Transactions on Information Systems*. DOI: 10.1145/3701228.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2022). MTEB: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Muennighoff, N., Tazi, N., *et al*. (2023). Mteb: Massive text embedding benchmark. `https://huggingface.co/spaces/mteb/leaderboard`.

Nishikawa, T., Hidaka, S., Yokoi, S., and Nakayama, H. (2022). Towards entity-enhanced RAG: Augmenting retrieval augmented generation with entity annotation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Paranhos, S. L., Tomazini, J., Oliveira, S., Junior, C. C., and de Oliveira, S. (2025). Preliminary ranking-based selection for optimized retriever configuration in rag systems. In *Anais do XL Simpósio Brasileiro de Bancos de Dados*, pages 725–738, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sbbd.2025.247492.

Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., and Shoham, Y. (2023). In-context retrieval-augmented language models. *Transactions on Machine Learning Research*.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Saad-Falcon, J., Khattab, O., Potts, C., and Zaharia, M. (2024). Ares: An automated evaluation framework for retrieval-augmented generation systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 338–354. DOI: 10.18653/v1/2024.naacl-long.20.

Safjan, K. (2023). From fixed-size to nlp chunking - a deep dive into text chunking techniques. Accessed: 2025-03-31.

Sakai, T. (2006). Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th Annual International ACM SIGIR Conference*, pages 525–532. ACM.

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. (2022). Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the International Conference on Neural Information Processing Systems*.

Wang, X., Dong, X., Zhang, F., Wang, L., and Xie, X. (2021). Kepler: A unified model for knowledge embedding and pre-trained language representation.

Zhong, Z., Liu, H., Cui, X., Zhang, X., and Qin, Z. (2025). Mix-of-granularity: Optimize the chunking granularity for retrieval-augmented generation.