Optimizing Efficiency in Multi-Stage Semantic Re-ranking Architectures

Artur M. A. Novais and Anna P. V. L. B. Moreira and Maria C. X. de Almeida
João P. C. Presa and **Fernando M. Federson** and **Savio S. T. de Oliveira**
Institute of Informatics – Federal University of Goiás (UFG)
{artur.matos, annapietra, mariaalmeida2}@discente.ufg.br
joaopaulopresa@gmail.com
{federson, savioteles}@ufg.br

Abstract

Semantic re-ranking architectures based on cross-encoders are essential for high-precision Information Retrieval (IR) in the legal domain, but they face a dilemma: their high computational latency renders large-scale applications challenging, particularly in resource-constrained environments. Traditional single-stage approaches force a choice between computational efficiency and ranking quality. This work presents an empirical evaluation of established cascade re-ranking architectures to optimize this balance through the adaptive application of off-the-shelf models of increasing complexity over progressively smaller sets of candidates. We validated the architecture on a corpus of 300,000 legal documents in Portuguese from the Court of Accounts of the State of Goiás (TCE-GO). Experiments demonstrate a 60.3% reduction in latency (from 11.75s to 4.66s per query) compared to the most precise single-stage baseline, with a marginal degradation of only 2 percentage points in R@avg and 0.0224 in MRR@avg. The results validate the semantic funnel as a computationally viable solution for semantic document-to-document search within the specific context of the TCE-GO repository, establishing a baseline for future transferability studies in broader Portuguese legal contexts.

1 Introduction

Information Retrieval (IR) in the legal domain is a crucial task for ensuring fairness and efficiency in case analysis, enabling the rapid retrieval of relevant jurisprudence within vast textual databases (Sansone and Sperlí, 2021). Natural Language Processing (NLP) has proven essential to the field, allowing search systems to comprehend the deep semantics of legal texts, going beyond simple keyword matching (Nguyen et al., 2025). Modern semantic search systems are frequently architected in two stages: retrieval, which selects a broad set of candidates, and re-ranking, which refines the order of this set to maximize precision at the top of the list (Nogueira et al., 2019).

The re-ranking stage presents a fundamental engineering dilemma: the balance between ranking quality and computational cost. The most effective models, based on cross-encoders that apply cross-attention between the query and the document, offer a much more refined capture of relevance (Capannini et al., 2016). However, this approach imposes a high latency cost, as operations cannot be pre-indexed and must be executed in real-time for each (query, document) pair, making its direct application unfeasible in large-scale databases (Zhang et al., 2023).

Traditional approaches to mitigate this latency problem generally rely on the application of a single re-ranking model (reranker) over a fixed subset of the Top-n documents returned by the retriever (Guo et al., 2024). This solution, however, forces a direct compromise: either a light and fast model is used, sacrificing final precision, or a heavy and precise model is used, limiting the number of documents that can be re-ranked and risking the discard of relevant candidates even before the refinement stage. Such approaches fail to optimize the balance between cost and performance, especially in domains such as the legal field, where precision in the top results is critical.

In this work, we empirically evaluate a cascade re-ranking architecture, applying established multi-stage techniques to optimize this balance in the context of legal data search. Our approach utilizes a progressive semantic funnel: (1) an initial retriever, based on a small bi-encoder model, selects a broad set of candidates; (2) a first low-latency reranker filters this set to a smaller subset; (3) a moderate-latency reranker refines the list again; and (4) a final high-performance, high-latency reranker processes only the most promising candidates to generate the final ranking. The central hypothesis

Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026)
Vol. 1, pages 562–571

is that, by applying computational power adaptively and concentrating the heaviest models only where they are most needed, we can achieve superior ranking quality (high R@avg and MRR@avg) with significantly lower latency than in single high-performance reranker approaches.

2 Related work

The proposed cascade re-ranking architecture lies at the intersection of three research areas within NLP. To contextualize and ground our results, this section reviews the literature informing the architecture of the proposed solution. First, we analyze the state-of-the-art in NLP architectures for Legal Information Retrieval (LIR), establishing how complex ranking structures are utilized in sensitive domains. Next, we delve into the core problem motivating this study: the trade-off between cost and performance in efficient re-ranking, investigating approaches that seek to optimize the latency of cross-encoders. Finally, we review recent work and benchmarks for the specific models comprising the semantic funnel, notably the BGE-Reranker and Qwen-Reranker families, to justify their selection for different stages of the cascade.

2.1 Legal Information Retrieval

Legal Information Retrieval (LIR) presents unique challenges that differentiate it from generic web search. Legal texts are dense, replete with specific terminology, and contain complex cross-references, rendering simple keyword matching insufficient. As highlighted by van Opijnen and Santos (2017), the very concept of "relevance" in the legal context is multifaceted and complex, requiring systems to understand the semantic intent behind a query rather than just lexical terms.

To address this gap in semantic understanding, the NLP community has developed Language Models (LMs) pre-trained specifically on large volumes of legal texts. Seminal works in this area include Legal-BERT, which demonstrated the superiority of LMs adapted to the legal domain in legal classification and information retrieval tasks (Chalkidis et al., 2020). In the Brazilian context, a fundamental milestone is Juru, a model trained specifically with legal content from Brazil, demonstrating a high capacity for understanding the nuances of Portuguese legalese (Junior et al., 2024). In parallel, benchmarks such as LexGLUE have been proposed to systematically evaluate and compare the performance of these models on tasks such as case classification, legal entity recognition, and legal sentence matching (Chalkidis et al., 2022).

While these advancements have largely resolved the challenge of domain understanding, practical implementation in large-scale search systems exposes the problem of efficiency. Many of these works focus on maximizing quality metrics such as F1, Accuracy, or MRR, often relying on single-stage re-ranking architectures with cross-encoders, such as Juru or Legal-BERT themselves. Although recent works have begun to explore multi-stage architectures, such as "Retrieve-Revise-Refine" (Nguyen et al., 2025), optimizing the balance between cost and performance in single-stage ranking architectures remains a challenge.

2.2 Efficient Multi-Stage Re-ranking Architectures

The literature clearly establishes a trade-off between quality and efficiency: more complex and computationally expensive ranking models, generally produce superior ranking quality but at a high latency cost (Sasazawa et al., 2023; Zheng et al., 2024). The reranker is often the component imposing the highest cost in an IR pipeline, as it must compare each query with multiple documents in a contextualized manner.

The most consolidated architectural solution in the literature to address this latency challenge is multi-stage or cascade ranking. The principle of this approach is to apply computational power progressively: an initial retriever (e.g., BM25 or bi-encoder) selects a broad set of candidates; one or more intermediate stages, using fast but less precise models, filter this list; and a final, slower, and more precise reranker (cross-encoder) is applied only to the subset of the most promising documents (Althammer et al., 2021; Zhu et al., 2022).

This funnel architecture has been empirically validated in various works. For instance, the BERT model was applied in a cascade architecture for passage retrieval, demonstrating that it is possible to achieve a robust balance between quality and latency by controlling how many candidates pass to the more expensive stages (Nogueira et al., 2019). More recent works expand this concept to the joint optimization of stages, with end-to-end learning of the cascade (Zheng et al., 2024).

Therefore, the semantic funnel proposal of this research is not the creation of a new method, but rather the empirical development of a cascade ar-

chitecture within the Portuguese legal context, utilizing the latest generation of rerankers (BGE and Qwen). This approach seeks to optimize the trade-off between cost and performance, reproducing the adaptive resource allocation behavior that has proven effective in multi-stage pipelines in general IR domains, but remains underexplored in the Brazilian legal field.

2.3 High-Performance Re-ranking Models: BGE and Qwen

BGE-M3 and its associated rerankers (e.g., BGE-Reranker-V2-M3) are widely recognized for their ability to handle multiple text granularities and for their high precision in retrieval benchmarks (Chen et al., 2024). For example, Guo et al. (2024) demonstrated that using BGE-Reranker in RAG pipelines improves the factual coherence of generated answers. Similarly, Yang et al. (2025) report consistent gains in Question Answering (Q&A) tasks when combining BGE-Reranker with dense retrievers.

The Qwen family of models has also excelled in Q&A and neural re-ranking tasks. Recent studies show that smaller-scale versions, such as Qwen-0.6B, can achieve competitive performance with lower costs compared to large-scale models, while larger variants, such as Qwen-4B, establish the state-of-the-art in semantic inference tasks (Bai et al., 2023). This modularity makes Qwen models particularly suitable for multi-stage architectures, where smaller models can act in early stages, and more robust versions are reserved for final refinement.

The existence of variants with different sizes allows for the construction of pipelines where each stage is calibrated according to computational cost and marginal precision gain (Sasazawa et al., 2023). Three central points become evident: LIR in Portuguese presents complex semantic challenges requiring specialized LMs; cascade ranking architectures are a robust and valid solution in generic IR to handle the latency-precision trade-off; and models such as BGE and Qwen comprise the current state-of-the-art in reranking models, offering scalable variations in cost and performance.

Our solution combines these three elements—LIR, multi-stage re-ranking, and state-of-the-art models (BGE and Qwen)—into a complete workflow, filling an important gap by presenting empirical results and a viable architectural proposal to optimize the balance between computational cost and precision in the domain of Portuguese legal case search.

3 Methodology and Datasets

This section describes the components of the proposed architecture, the corpus used for experimental validation, and the evaluation procedures. The methodology was structured to allow for a systematic comparison between single-stage architectures and the proposed cascade re-ranking approach.

3.1 Corpus and Test Set

The experiments were conducted using a corpus composed of 300,000 textual documents from 16,000 lawsuits of the Court of Accounts of the State of Goiás (TCE-GO). This corpus represents a significant sample of the volume and linguistic diversity found in real-world IR systems in the Brazilian legal domain. Each lawsuit contains multiple documents (procedural pieces, dispatches, opinions, decisions, technical statements, among others), resulting in a document granularity that reflects the actual structure of search systems used by courts. The texts were pre-processed for the removal of non-textual elements (automated headers, digital signatures, formatting metadata) and segmented into document units consistent with the application, preserving the semantic integrity of each procedural piece.

As previously highlighted, the practical application motivating this work is the similarity search for legal cases, a scenario where the user possesses a reference document and wishes to find similar documents in the database. Unlike traditional search systems based on short text queries, the system operates on full documents as input, seeking to identify other semantically related documents in the corpus. To evaluate the effectiveness of retrieval systems in this context, we constructed a test set composed of 250 document-summary pairs. For each selected document, a summary containing its essential semantics was generated, capturing the central elements of the original document and maintaining semantic coherence, but with a distinct textual formulation.

A fundamental premise of this work is that a document should have no other item more "similar" to it than its own summary. Thus, for each of the 250 documents in the validation set, the "relevant pair" is defined as the original document and its respective summary. This approach establishes an

objective and rigorous relevance criterion: an effective retrieval system must be able to position the summary of a document in the top positions when the full document is used as the query. This methodology offers important advantages, including: it eliminates the need for extensive manual annotation of multiple relevant documents for each query; it defines relevance clearly and unambiguously; and it reflects a real-world use case in legal systems, where professionals frequently search for precedents or cases similar to a specific case they have in hand. However, we acknowledge that this document-summary setup does not cover all nuances of real-world legal search, such as short question-answering queries (e.g., "prescription in bidding processes") or relevance dependent on doctrinal interpretations and indirect precedents. Since this is a document-to-document similarity system, the term "query" refers, in this work, to the full document used as input for the search, and not to a question or short phrase, as is common in Q&A systems.

3.2 System Architecture

The proposed architecture is organized as a progressive semantic funnel, composed of four sequential stages designed to balance efficiency and precision. An overview of this processing flow is presented in Figure 1.

The foundation of the retrieval stage lies in the prior construction of a vector index using the FAISS (*Facebook AI Similarity Search*) library. Consistent with the tool's architecture, this index stores the latent representations of all 300,000 documents in the corpus, structuring them to allow for efficient search operations based on Euclidean distances or dot products. At query time, the system performs a similarity scan on the index to retrieve the most relevant candidates. This process presents significantly reduced latency compared to rerankers, as it leverages the bi-encoder architecture that decouples encoding: since document vectors are pre-computed offline, the online cost is limited to fast algebraic operations, avoiding the heavy cross-attention inference of subsequent models. Figure 1 illustrates this flow with exemplary cutoff values (Top-100, Top-20, and Top-10); however, it is emphasized that the parameter N (defined as the number of top-ranked candidates retrieved in the initial stage to be processed by the subsequent re-ranking models) and subsequent cutoffs are experimental variables (tested, for example, with $N \in \{20, 30, 50\}$) adjustable according to recall needs. The Qwen3-Embedding-0.6B model was used to generate the embeddings that populate this index and encode the user query.

After initial retrieval, candidates proceed to the re-ranking stage visualized in the center and right of Figure 1. Technically, all re-ranking models (BGE and Qwen family) operate as *cross-encoders*. Given a pair (query document q, candidate document d), the model processes the concatenation $[q; d]$ and generates a relevance score $s(q, d) \in \mathbb{R}$. The process occurs in three steps: (1) the sequence `[CLS] query_document [SEP] candidate_document [SEP]` is tokenized; (2) a Transformer encoder applies cross-attention between the query and the candidate; and (3) the representation of the `[CLS]` token is linearly projected to a scalar score. This approach, distinct from the initial vector search, allows for capturing deep semantic relations essential for the system's final precision.

The logic of progressively reducing the number of candidates ($N > N_1 > N_2 > N_3$) is grounded in the principle of efficient resource allocation: more expensive models (such as Qwen 4B) should process only documents where their semantic disambiguation capacity is strictly necessary. This stage applies models of increasing complexity over progressively smaller subsets of data:

1. **Stage 1 (low latency):** the first re-ranking block utilizes the BGE-Reranker-V2-M3. As a lightweight model with a smaller context window, its function is to rapidly filter the raw list from the retriever, eliminating irrelevant candidates and prioritizing recall.

2. **Stage 2 (moderate latency):** relevant candidates are refined by the Qwen3-Reranker-0.6B. This medium-sized model expands the context, allowing for a more detailed analysis.

3. **Stage 3 (high latency):** the final stage applies the Qwen3-Reranker-4B over the most restricted set. Being the most robust model, it concentrates the highest computational cost only on the most promising candidates to generate the definitive ranking.

3.3 Experimental Setup

To validate the central hypothesis of this work, we compared three types of architectures: (1) *Baseline* (*Retriever-only*) without re-ranking; (2) *Single-model*, retriever plus a single reranker applied over

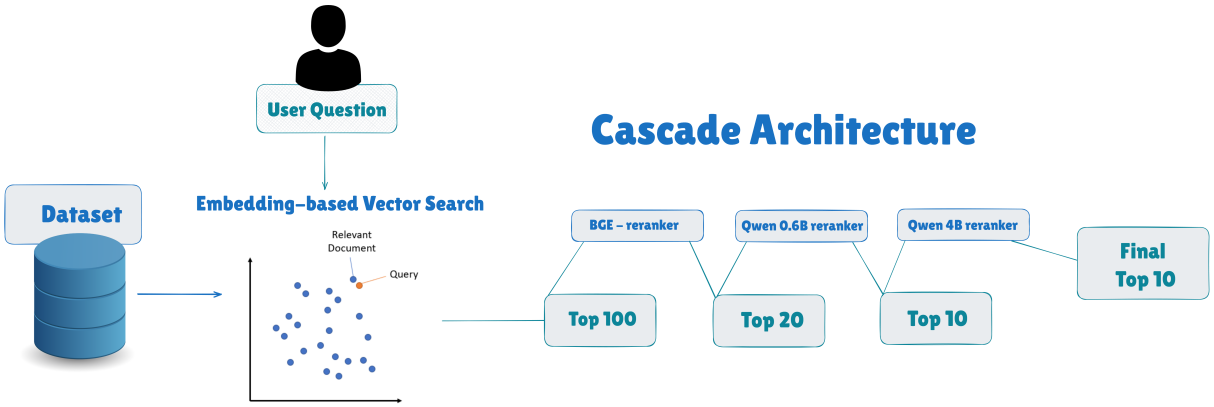

Figure 1: Overview of the cascade re-ranking architecture, illustrating the flow from vector search to final refinement.

Top-N documents, testing each reranker individually (BGE, Qwen-0.6B, Qwen-4B) with different values of N and maximum context lengths; (3) *Cascade* that uses a retriever plus multiple rerankers in cascade, with different configurations of N, including two-stage configurations and three-stage cascades such as BGE (*Top-30*), followed by Qwen-0.6B (*Top-20*), and finally Qwen-4B (*Top-10*).

Validating the architecture requires simultaneously evaluating effectiveness (ranking quality) and efficiency (computational cost). To quantify ranking quality, we used *Recall@k* ($R@k$) and *Mean Reciprocal Rank@k* ($MRR@k$), adapted to the context of document similarity search. $R@k$ measures whether the relevant document (the corresponding summary) appears among the first k positions of the ranking, being binary (0 or 1) for each test. The final metric is the average of the 250 pairs: $R@k_{\text{final}} = \frac{1}{250}\sum R@k_i$. The values of k used were $\{1, 3, 5, 10\}$, with $R@1$ being particularly critical as it indicates whether the summary was positioned as the most similar document to the original. $MRR@k$ measures how "quickly" the relevant document appears in the ranking: $MRR@k = \frac{1}{250}\sum \frac{1}{\min(\text{rank}_i, k+1)}$, where rank_i is the position of the summary in the ranking for the i-th query document. This metric captures not only whether the summary was found but also the quality of its position in the ranking. To facilitate the general comparison between systems, we report the aggregated averages $R@\text{avg}$ (arithmetic mean of $R@1$, $R@3$, $R@5$, and $R@10$) and $MRR@\text{avg}$ (arithmetic mean of $MRR@1$, $MRR@3$, $MRR@5$, and $MRR@10$), which provide a holistic view of system performance at different ranking depths.

Efficiency was measured through latency (*Average Time*), defined as the total time (in seconds) required to process a single query, from initial retrieval to final ranking. All measurements were performed on an NVIDIA A100 GPU (80GB), with batch size = 1 (simulating sequential queries), using the Hugging Face *Transformers* library with *float16* precision. Each query was executed multiple times, reporting the median execution time to mitigate variations due to system overhead. The goal of the analysis is to identify configurations that optimize the trade-off between effectiveness and efficiency, offering quality close to that of the most precise single-stage systems but with substantially reduced latency.

All models were obtained from the Hugging Face Model Hub: Qwen3-Embedding-0.6B (Alibaba-NLP/gte-Qwen3-0.6B-instruct), BGE-Reranker-V2-M3 (BAAI/bge-reranker-v2-m3), Qwen3-Reranker-0.6B (Alibaba-NLP/gte-Qwen3-0.6B-reranker), and Qwen3-Reranker-4B (Alibaba-NLP/gte-Qwen3-4B-reranker).

The FAISS index was built using the `IndexFlatIP` (*Inner Product*) type, suitable for normalized embeddings. The cascade implementation was developed in Python, utilizing the `transformers`, `faiss-gpu`, and `torch` libraries.

4 Results

To validate the hypothesis that a cascade architecture optimizes the balance between effectiveness and efficiency, the experiments were structured in three stages: (1) single-stage *baselines* (isolated *retriever* and *rerankers*); (2) *two-stage* architectures; and (3) complete three-stage cascades.

4.1 Single-Stage Reranking

Table 1 establishes the baseline performance of the system operating exclusively with the dense retrieval stage (Qwen3-Embedding-0.6B). The analysis of these metrics reveals the typical behavior of *bi-encoder* models: high overall recall capacity at the expense of fine-grained precision at the top of the list.

It is observed that the system achieves an R@10 of 97.2%, indicating that in almost all queries (243 out of 250 pairs), the relevant document was retrieved and positioned among the top 10 candidates. This result validates the effectiveness of the FAISS index and the embedding model as a robust initial filter, ensuring that the correct documents are available for subsequent stages.

However, there is a notable degradation when observing R@1 (81.2%). There is a 16 percentage point gap between the model's ability to find the document (R@10) and its ability to correctly position it as the first result. This suggests that while the dot product of dense vectors captures global semantic similarity well, it fails to distinguish subtle nuances necessary for perfect ordering in ambiguous cases. The MRR@avg of 0.8505 corroborates this analysis: the system is competent but leaves a clear margin for optimization that justifies the introduction of re-ranking architectures based on *cross-encoders* to refine this final ordering.

System	Recall	MRR
Retriever (baseline)	R@1: 81.2%	MRR@1: 0.8120
	R@3: 92.0%	MRR@3: 0.8587
	R@5: 94.4%	MRR@5: 0.8639
	R@10: 97.2%	MRR@10: 0.8675
	R@avg: 91.2%	**MRR@avg: 0.8505**

Table 1: Retriever baseline (without *reranker*). Aggregated metrics on the full set.

Table 2 explores the performance of single-stage reranking architectures, revealing the central trade-off between ranking quality and computational latency. The Qwen3-Reranker-4B (MaxLen=8192, Top-in=20) establishes the performance ceiling for this dataset, achieving the highest precision across all metrics (R@1=89.20%, MRR@avg=0.9124). Compared to the retriever baseline, this represents a significant gain of 8 percentage points in R@1. However, this accuracy comes at a prohibitive cost: the model requires an average of 11.75 seconds per query, making it unfeasible for real-time applications with high throughput.

Conversely, the BGE-Reranker-v2-m3 highlights the limitations of lightweight models in this complex legal domain. Although it is faster than other models (processing 50 candidates in just 0.90 seconds) its performance is notably inferior to the retriever baseline (R@avg=82.60% vs. 91.20%). This indicates that, when used in isolation, the BGE model acts as a bottleneck, degrading the quality of the initial candidate list provided by the dense retriever.

The intermediate model, Qwen3-Reranker-0.6B, offers a middle ground but still imposes significant latency (approx. 5s for Top-50). Interestingly, increasing the context length from 4096 to 8192 tokens for the 0.6B model resulted in a performance drop (R@1 fell from 86.40% to 84.40%) and doubled the execution time (18.02s), suggesting that smaller models may struggle to effectively attend to very long contexts without introducing noise. These findings underscore the necessity of a cascade architecture that leverages the speed of BGE for initial filtering and the precision of Qwen-4B for the final selection.

4.2 Cascade Architectures

To overcome the latency limitations of single-stage models, we investigated composite architectures. The following analysis details performance evolution through two-stage configurations, three-stage cascades, and the final consolidated comparison.

4.2.1 Two-Stage Configurations

Table 3 evaluates pipelines composed of two rerankers from the Qwen family, specifically chaining the medium model (Qwen3-0.6B) feeding into the large model (Qwen3-4B). The rationale behind these experiments is to utilize the 0.6B model as a "gatekeeper" that is sufficiently semantically aware to filter out non-relevant documents that might confuse a simpler model, yet significantly faster than the 4B model. The results indicate that the configuration chaining Qwen3-0.6B (Top=30) into Qwen3-4B (Top=10) offers the most balanced performance.

By reducing the input pool from 30 to 10 candidates before the final stage, the system achieves an MRR@avg of 0.8881 while reducing latency to 6.15s, which represents a 47.6% reduction compared to running the Qwen-4B model alone (11.75s). Furthermore, comparing the first and second rows of the table reveals that increasing the intermediate output size (Top-20 vs. Top-10) yields negligible accuracy gains but increases processing

Table 2: Results of single-stage baselines (*Single-model*). Compares Qwen and BGE with different configurations.

Model	MaxLen	Top-in	R@1	R@3	R@5	R@10	MRR@1	MRR@3	MRR@5	MRR@10	R@avg	MRR@avg	Time (s)
Qwen3-Reranker-4B	8192	20	**89.20%**	92.80%	**96.00%**	97.60%	**0.8956**	**0.9123**	**0.9195**	**0.9221**	**93.90%**	**0.9124**	11.7483
Qwen3-Reranker-0.6B	4096	100	86.40%	**93.20%**	**96.00%**	98.00%	0.8675	0.8983	0.9045	0.9073	93.40%	0.8944	9.9644
Qwen3-Reranker-0.6B	4096	50	86.00%	**93.20%**	95.60%	98.00%	0.8635	0.8963	0.9017	0.9050	93.20%	0.8916	5.0795
Qwen3-Reranker-0.6B	8192	100	84.40%	92.40%	**96.00%**	**98.80%**	0.8474	0.8842	0.8924	0.8963	92.90%	0.8801	18.0230
BGE-reranker-v2-m3	1024	50	71.60%	80.40%	86.40%	92.00%	0.7189	0.7590	0.7727	0.7799	82.60%	0.7576	**0.9042**
BGE-reranker-v2-m3	2048	50	68.00%	82.40%	87.60%	92.00%	0.6827	0.7450	0.7568	0.7622	82.50%	0.7367	1.6672
BGE-reranker-v2-m3	1024	100	69.60%	79.60%	82.40%	90.00%	0.6988	0.7436	0.7503	0.7604	80.40%	0.7383	1.7721
BGE-reranker-v2-m3	8192	100	48.80%	66.40%	72.00%	82.00%	0.4900	0.5683	0.5813	0.5950	67.30%	0.5587	12.0913

time significantly (to 9.08s), validating the premise that the computationally expensive final reranker should only process the absolute most promising candidates.

4.2.2 Three-Stage Cascades

Table 4 introduces the full "semantic funnel" by prepending the BGE-Reranker-v2-m3 as the first stage ($S1$). Although BGE demonstrated inferior performance in isolation (as detailed in Table 2), its inclusion in the cascade is strategic due to its ultra-low latency. The configuration labeled "Cascade (fast)" — comprising BGE (Top=30), followed by Qwen-0.6B (Top=20), and finally Qwen-4B (Top=10) — reaches the optimal efficiency point of 4.66s per query.

By employing BGE to perform a "coarse sort" on a larger initial set, the system can aggressively prune clearly irrelevant documents. This protects the subsequent Qwen models from processing noise, allowing them to focus their attention mechanisms on hard negatives. Notably, the results show that adding a third stage actually decreases total processing time compared to the two-stage approach (4.66s vs 6.15s). This counter-intuitive result occurs because the initial BGE stage filters the list so efficiently that the input size for the subsequent, more expensive Qwen-0.6B stage is smaller and cleaner than in the two-stage setup.

4.2.3 Synthesis: Effectiveness-Efficiency Trade-off Analysis

Table 5 provides the consolidated view of the best-performing configurations from each architectural approach, effectively synthesizing the core contribution of this study. The comparison highlights clear trade-off boundaries between the different strategies. The configuration labeled "Cascade (fast)" achieves the lowest latency among all reranking setups at 4.66s, representing a 60.3% speedup compared to the state-of-the-art Single-model. Notably, this three-stage configuration outperforms even the two-stage lightweight configuration (5.19s), demonstrating that adding a third, ultra-fast stage (BGE) to prune the list early is more computationally efficient than feeding a larger candidate list directly to the intermediate Qwen-0.6B model.

Conversely, the Single-model Qwen-4B remains the upper bound for precision with an MRR@avg of 0.9124, serving as the theoretical ceiling for retrieval quality. However, its computational cost of 11.75s renders it impractical for interactive search sessions where throughput and response time are critical. In this context, the "Cascade (complete)" configuration emerges as the most robust solution for production environments. By slightly increasing the initial pool ($N = 50$), it recovers precision to an MRR@avg of 0.8955—a level very close to the single-model ceiling—while maintaining a latency of 4.86s, which is less than half the time required by the heavy baseline. Ultimately, the data demonstrate that the proposed architecture allows the system to operate much closer to the retriever's speed while delivering ranking quality comparable to the most expensive cross-encoders.

4.3 Results Discussion

We emphasize that established Portuguese-specific models, such as Juru (Junior et al., 2024), were not included in the direct reranking comparison. Juru is based on the BERT architecture with a 512-token limit, which renders it unsuitable for the full-document similarity task proposed here without extensive truncation strategies that would compromise the semantic comparison of long legal pieces. The selected BGE and Qwen models were chosen specifically for their ability to handle extended contexts (up to 8k and 32k tokens).

Although the performance margins between the complete cascade and the single-stage baseline are narrow (approx. 2 p.p.), the primary contribution lies in the latency reduction. We acknowledge that statistical significance tests were not applied to these differences, and future work should rigorously validate whether these quality gaps are statistically discernible.

Table 3: Comparison between *two-stage* configurations (Qwen3-0.6B → Qwen3-4B), with **MaxLen=**4096.

Pipeline (S1 → S2)	R@1	R@3	R@5	R@10	MRR@1	MRR@3	MRR@5	MRR@10	R@avg	MRR@avg	Avg. Time (s)
Qwen3-0.6B (Top=20) → Qwen3-4B (Top=10)	**85.60%**	92.00%	**95.60%**	97.60%	**0.8594**	0.8889	0.8973	0.9001	92.70%	0.8864	**5.19**
Qwen3-0.6B (Top=30) → Qwen3-4B (Top=10)	**85.60%**	**92.80%**	**95.60%**	**98.40%**	**0.8594**	**0.8922**	**0.8985**	**0.9024**	**93.10%**	**0.8881**	6.15
Qwen3-0.6B (Top=30) → Qwen3-4B (Top=20)	**85.60%**	**92.80%**	**95.60%**	**98.40%**	**0.8594**	**0.8922**	**0.8985**	**0.9024**	**93.10%**	**0.8881**	9.08

Table 4: Comparison between cascade configurations (BGE → Qwen → Qwen).

Pipeline (S1 → S2 → S3)	R@1	R@3	R@5	R@10	MRR@1	MRR@3	MRR@5	MRR@10	Avg. Time (s)
BGE-v2-m3 (Top=50, MaxLen=1024) → Qwen3-0.6B (Top=20, MaxLen=4096) → Qwen3-4B (Top=10, MaxLen=4096)	**87.20%**	**92.80%**	**94.00%**	94.00%	**0.8755**	**0.9003**	**0.9031**	**0.9031**	4.86
BGE-v2-m3 (Top=30, MaxLen=1024) → Qwen3-0.6B (Top=20, MaxLen=4096) → Qwen3-4B (Top=10, MaxLen=4096)	86.40%	92.40%	**94.00%**	**94.80%**	0.8675	0.8949	0.8981	0.8994	**4.66**
BGE-v2-m3 (Top=100, MaxLen=1024) → Qwen3-0.6B (Top=20, MaxLen=4096) → Qwen3-4B (Top=10, MaxLen=4096)	84.40%	91.60%	92.40%	92.40%	0.8474	0.8815	0.8833	0.8833	5.38

Table 5: Final comparison between the best configurations (Quality vs Cost). **Bold** = best per column (for **Time**, lower is better). †Retriever time is not included in the *reranking* cost.

Type	Configuration (explicit parameters)	R@avg	MRR@avg	Avg. Time (s)
Baseline (Retriever)	No reranker (retriever `top_ids` list)	91.20%	0.8505	—†
Single-model (Qwen)	Qwen3-Reranker-4B [Top-in=20, MaxLen=8192]	**93.90%**	**0.9124**	11.75
Two-stage (Qwen → Qwen)	S1: Qwen3-0.6B [Top-in=30, MaxLen=4096] → S2: Qwen3-4B [Top-in=10, MaxLen=4096]	93.10%	0.8881	6.15
Two-stage (Qwen → Qwen, lightweight)	S1: Qwen3-0.6B [Top-in=20, MaxLen=4096] → S2: Qwen3-4B [Top-in=10, MaxLen=4096]	92.70%	0.8864	5.19
Cascade (BGE → Qwen → Qwen, fast)	S1: BGE-v2-m3 [Top-in=30, MaxLen=1024] → S2: Qwen3-0.6B [Top-in=20, MaxLen=4096] → S3: Qwen3-4B [Top-in=10, MaxLen=4096]	91.90%	0.8900	**4.66**
Cascade (BGE → Qwen → Qwen, complete)	S1: BGE-v2-m3 [Top-in=50, MaxLen=1024] → S2: Qwen3-0.6B [Top-in=20, MaxLen=4096] → S3: Qwen3-4B [Top-in=10, MaxLen=4096]	92.00%	0.8955	4.86

The experimental results validate the "progressive semantic funnel" as a distinct and effective architectural pattern for legal information retrieval, offering insights that extend beyond simple model ensembling. The results underscore the economic value of early pruning in computational pipelines. The counter-intuitive finding that a three-stage cascade can be faster than a two-stage approach (4.66s versus 6.15s) highlights the critical role of the initial filtering stage. The BGE model acts as a highly efficient filter, removing easy negatives at a negligible cost. This mechanism protects the downstream, computationally expensive Qwen models from processing irrelevant documents, thereby maximizing the return on computational investment regarding floating-point operations per relevant document found.

The reduction in latency acts as a primary enabler for user experience in real-world applications. In the context of the Brazilian legal system, where databases grow exponentially, a query latency of nearly 12 seconds—as seen in the single-model approach—is often unacceptable for interactive research tools used by legal professionals. By bringing this down to the 4-second range without significantly sacrificing semantic understanding, the cascade architecture crosses the threshold from theoretical viability to practical implementation, allowing for a fluid search experience.

Finally, the robustness of the retriever-only baseline is noteworthy, achieving an MRR@avg of 0.8505. This suggests that for applications where sub-second latency is a strict requirement, dense retrieval alone serves as a competent solution. However, for the specific task of legal precedent discovery, where finding the exact most relevant case is critical to the legal argument, the gain of approximately 4 to 6 percentage points in MRR provided by the cascade justifies the additional computational overhead, provided it remains within the efficient bounds established by the proposed architecture.

5 Conclusion

This work addressed an important challenge of semantic Information Retrieval in the legal domain: the *trade-off* between *ranking* quality and computational cost in systems based on *cross-encoders*. Traditional single-stage approaches force a binary choice between fast but imprecise models, and precise but computationally infeasible models. A multi-stage re-ranking architecture (progressive semantic funnel) was proposed and empirically validated to optimize this balance through the adaptive allocation of computational resources.

The main contribution is to empirically demonstrate that progressive semantic funnels constitute a viable architectural solution for large-scale legal search systems. By concentrating computationally expensive models only on the most promising candidates, the cascade maintains high *ranking* quality with substantially reduced cost, enabling implementation in production environments.

While our experimental evaluation demonstrates the viability of the semantic funnel, it is restricted to documents from a single institution (TCE-GO). Future work must validate the transferability of these findings to other branches of law (e.g., Civil or Penal), courts with different documentary standards, and other Portuguese-speaking jurisdictions.

Furthermore, our evaluation utilized generalist dense representations (BGE and Qwen families). Comparing this architecture with domain-specific models like Juru and learning-to-rank (LTR) approaches remains a critical next step. Finally, recognizing that our latency tests were performed on a high-performance NVIDIA A100 GPU (80GB), future research should investigate the architecture's behavior on the more modest hardware infrastructures commonly found in Brazilian public institutions to ensure broad practical viability.

Limitations

While this study demonstrates the efficiency of the progressive semantic funnel, it presents several limitations. First, the evaluation is restricted to a single institutional corpus (TCE-GO) and relies on a specific document-to-document similarity task based on document-summary pairs. Consequently, the findings may not fully generalize to other legal branches, different documentary standards, or scenarios involving short, complex user queries, such as legal question-answering.

Second, our latency experiments were conducted on a high-end NVIDIA A100 GPU (80GB). The architecture's practical viability on the more modest hardware infrastructures typically found in Brazilian public institutions remains to be investigated. Finally, domain-specific models for Brazilian Portuguese, such as Juru, were excluded from the direct reranking comparison due to their strict 512-token limits, which are unsuitable for long legal texts without extensive truncation. Furthermore, while the latency reductions are substantial, the narrow performance margins between the cascade and the single-stage baseline lack statistical significance testing, an aspect that should be rigorously addressed in future work.

Acknowledgments

This work has been supported by TCE-GO. This work was supported by the National Institute of Science and Technology (INCT) in Responsible Artificial Intelligence for Computational Linguistics and Information Treatment and Dissemination (TILD-IAR) grant number 408490/2024-1.

References

S Althammer, A Askari, S Verberne, and AH Hanbury. 2021. Dossier@ coliee 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. *Proceedings of the eighth international competition on legal information extraction/entailment (COLIEE 2021)*, pages 8–14.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

G. Capannini, C. Lucchese, F. M. Nardini, S. Orlando, R. Perego, and N. Tonellotto. 2016. Quality versus efficiency in document scoring with learning-to-rank models. *Information Processing & Management*, 52(5):902–918.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2910, Online. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Jun Guo, Bojian Chen, Zhichao Zhao, Jindong He, Shichun Chen, Donglan Hu, and Hao Pan. 2024. Bkrag: A bge reranker rag for similarity analysis of power project requirements. In *Proceedings of the 2024 6th International Conference on Pattern Recognition and Intelligent Systems*, PRIS '24, page 14–20, New York, NY, USA. Association for Computing Machinery.

Roseval Malaquias Junior, Ramon Pires, Roseli Romero, and Rodrigo Nogueira. 2024. Juru: Legal brazilian large language model from reputable sources. *arXiv preprint arXiv:2403.18140*.

C. Nguyen, P. M. Nguyen, and N. Le. 2025. Retrieve–Revise–Refine: A novel framework for retrieval of concise entailing legal article set. *Information Processing & Management*, 62:103949.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with BERT. *ArXiv preprint arXiv:1910.14424*.

C. Sansone and G. Sperlí. 2021. Legal Information Retrieval systems: State-of-the-art and open issues. *Information Systems*, 101:101967.

Yoshihiko Sasazawa, Kazuya Yokote, Osamu Imaichi, and Yusuke Sogawa. 2023. Text Retrieval with Multi-Stage Re-Ranking Models. *ArXiv preprint arXiv:2311.07994*.

M. van Opijnen and C. Santos. 2017. On the concept of relevance in legal information retrieval. *Artificial Intelligence and Law*, 25(1):65–87.

Te-Lun Yang, Jyi-Shane Liu, Yuen-Hsien Tseng, and Jyh-Shing Roger Jang. 2025. Knowledge Retrieval Based on Generative AI. *Preprint*, arXiv:2501.04635.

Qin Zhang, Shangsi Chen, Dongkuan Xu, Qingqing Cao, Xiaojun Chen, Trevor Cohn, and Meng Fang. 2023. A survey for efficient open domain question answering. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 14447–14465.

K. Zheng, H. Zhao, R. Huang, B. Zhang, N. Mou, Y. Niu, Y. Song, H. Wang, and K. Gai. 2024. Full Stage Learning to Rank: A Unified Framework for Multi-Stage Systems. In *Proceedings of the ACM on Web Conference 2024*.

J. Zhu, X. Luo, and J. Wu. 2022. A BERT-Based Two-Stage Ranking Method for Legal Case Retrieval. In *Challenge of AI in Law (CAIL 2021)*, pages 534–546. Springer.