Análise Comparativa de Métodos de Recuperação de Informação no Contexto Jurídico Brasileiro

Pedro Henrique Honorio Saito[1][DRE: 122149392]

[1] Universidade Federal do Rio de Janeiro, Ilha do Fundão, Rio de Janeiro

Abstract. Sistemas de recuperação de informação jurídica lidam com desafios como linguagem técnica e alto volume de dados. Este estudo apresenta uma análise comparativa de abordagens lexicais, semânticas e híbridas aplicadas à jurisprudência brasileira, utilizando o *dataset* **JurisTCU** que contempla 16.045 documentos do Tribunal de Contas da União e 150 consultas com julgamentos de relevância. Avaliei: (1) modelos esparsos como variações do BM25 (*Lucene*, *BM25L*, *BM25+*, *ATIRE*, *BMX* e *Pyserini* com RM3); (2) Modelos densos baseados em *sentence embeddings* (GTE, Jina, Qwen, Gemma e variantes pré-treinadas); e (3) métodos de busca híbrido com *reranking* e algoritmos de fusão como *Reciprocal Rank Fusion* (RRF), *MNZ*, *WMNZ*, *WSUM*, *TM2C2*, dentre outros. Resultados indicam que a combinação de sinais lexicais e semânticos supera abordagens exclusivamente de um dos tipos em coleções reais de decisões judiciais.

Keywords: Recuperação de Informação · Busca Híbrida · Reranking · Reciprocal Rank Fusion · Jurisprudência Brasileira · Dense Retrievers · Embeddings · BM25

1 Introdução

Legal Information Retrieval (LIR) foca em métodos de busca aplicados a documentos jurídicos como leis, jurisprudências, acórdãos e processos. A recuperação de informação em domínios especializados, como o jurídico, apresenta desafios significativos devido à linguagem técnica, ao grande volume de documentos e à necessidade de capturar nuances semânticas complexas.

Com o crescimento contínuo da jurisprudência, modelos esparsos como o BM25 permanecem relevantes por serem eficientes e interpretáveis em cenários *out-of-domains* e consultas com termos léxicos específicos. No entanto, modelos de linguagem permitem o uso de *embeddings*, que capturam relações semânticas para além da correspondência exata dos termos.

O objetivo deste trabalho é realizar uma análise comparativa entre diferentes abordagens de busca no contexto de LIR em português. As abordagens avaliadas incluem:

- Modelos esparsos (variantes do BM25).
- Modelos densos baseados em *sentence embeddings*.
- Pipeline de duas etapas com reranqueadores.
- Algoritmos de fusão de busca híbrida e estratégias de normalização.

O resultado esperado é que a combinação de sinais lexicais e semânticos supere métodos puramente léxicos ou semânticos em coleções reais de decisões judiciais.

2 Fundamentação Teórica

2.1 BM25 e Variantes

O BM25 é uma função de ranqueamento que avalia a pertinência de um documento considerando a frequência dos termos na consulta, a raridade deles na coleção e a normalização pelo tamanho do documento. Diversas variantes foram propostas para melhorar o desempenho em diferentes cenários:

BM25 Lucene

Implementação padrão com suavização no cálculo do IDF e fator de normalização TF. A fórmula incorpora um termo de suavização (+1) no numerador do IDF para evitar valores negativos.

$$\text{BM25}_{lucene}(q, D) = \sum_{t \in q \cap D} \left[\log\left(1 + \frac{N - n_i + 0,5}{n_i + 0,5}\right) \cdot \frac{f_{t,D}(k_1 + 1)}{f_{t,D} + k_1 \cdot \left(1 - b + b\frac{|D|}{\text{avgdl}}\right)} \right]$$

De modo geral, é uma abordagem bem conservadora.

BM25L

Variante que modifica o componente TF para lidar melhor com documentos longos, introduzindo um parâmetro δ que ajusta a penalização por comprimento do documento

$$\text{BM25L}(q, D) = \sum_{t \in q \cap D} \left[\log\left(\frac{N - n_i + 0,5}{n_i + 0,5}\right) \cdot \text{TF}'_{t,D} \right]$$

$$\text{onde } \text{TF}'_{t,D} = \begin{cases} \frac{(k_1+1)\cdot\left(\frac{c}{\text{norm}}+\delta\right)}{k_1+\left(\frac{c}{\text{norm}}+\delta\right)} & \text{se } f_{t,D} > 0 \\ 0 & c.c. \end{cases}$$

O termo $\frac{c}{\text{norm}}$ é o TF normalizado pelo comprimento do documento:

$$\frac{c}{\text{norm}} = \frac{c_{t,D}}{1 - b + b\frac{|D|}{\text{avgdl}}}$$

O BM25 penaliza excessivamente documentos muito longos: A normalização faz $\frac{c}{\text{norm}}$ tender a zero, levando o TF a valores que praticamente ignoram a ocorrência do termo. Assim, documentos extensos deixam de ser distinguidos de documentos sem o termo.

O BM25L introduz um deslocamento $\delta > 0$, garantindo um limite inferior positivo para o TF sempre que há ocorrência. Isso preserva a separação entre presença e ausência do termo e evita o colapso do *score* em documentos longos.

A função ajustada mantém as propriedades essenciais do BM25, adicionando outras: TF nulo sem ocorrência, crescimento monotônico com a frequência e mínimo assintótico significativo quando o termo está presente.

BM25+ e ATIRE

As variantes BM25+ e ATIRE partem do mesmo princípio do BM25L de tratar documentos mais longos, porém com algumas variações. O BM25+ adiciona um *offset* positivo $\delta \times \text{IDF}(t)$ ao *score* final, garantindo que documentos com correspondência de termos sempre recebam *score* positivo. Por outro lado, a variante ATIRE modifica o cálculo da normalização por comprimento do documento, invertendo a relação entre o tamanho médio dos documentos (avgdl) e o tamanho do documento ($|D|$) no denominador do TF.

A função de ranqueamento de ambas as variantes está dada abaixo:

$$\text{BM25+}(q, D) = \text{OkapiBM25} + \delta \cdot \text{IDF}(t)$$

$$\text{BM25}_{\text{ATIRE}}(q, D) = \sum_{t \in q \cap D} \left[\text{IDF}(t) \cdot \frac{f_{t,D}(k_1 + 1)}{f_t, D + k_1 \cdot \left(b + (1 - b)\frac{avgdl}{|D|}\right)} \right]$$

Ambas as variantes ajustam o BM25 para reduzir a penalização de documentos longos e estabilizar o score, mantendo a estrutura básica do modelo.

BMX

Variante recente que incorpora entropia ponderada e componentes semânticos, adicionando termos para expansão de consulta (E) e similaridade semântica (S) com parâmetros α e β.

$$\text{BMX}(q, D) = \sum_{t \in q \cap D}^{m} \text{IDF}(q_i) \cdot \frac{f_{q,D} \cdot (\alpha + 1)}{f_{q,D} + \alpha \cdot \frac{|D|}{avgdl} + \alpha \cdot E} + \beta \cdot E(q_i) \cdot S(q, D)$$

Os parâmetros α e β controlam, respectivamente, a influência da componente lexical normalizada e o peso d parcela semântica de expansão de consulta, equilibrando a contribuição de E e S no *score* final.

Pyserini com RM3

Além das variantes implementadas diretamente, este trabalho também avalia o BM25 com RM3 por meio da biblioteca `Pyserini`, incorporando *pseudo-relevance feedback* como uma configuração esparsa adicional para comparação com os demais modelos.

2.2 Embeddings Contextuais

Modelos como BERT produzem token-level embeddings e não são otimizados para medir similaridade entre sentenças. Para resolver essa limitação, foi desenvolvido o Sentence-BERT (SBERT) com objetivos de treino específicos:

- **Cross-Entropy Loss (NLI)**: Aprende relações entre pares de sentenças classificando-as como implicação (entailment), sem relações lógicas (neutral) ou contradição (contradiction).
- **Cosine Similarity Loss (STS)**: O modelo recebe duas sentenças e prevê um valor de similaridade seguindo julgamento humano.
- **Multiple Negative Ranking Loss**: Cada item do batch de treinamento se torna um negativo para todos os outros, exceto seu par positivo, aumentando a eficiência do treino.
- **Matryoshka Learning**: Técnica que força um embedding maior conter múltiplas representações aninhadas de tamanhos fixos, permitindo flexibilidade na escolha da dimensionalidade sem necessidade de retreinamento.
- **Aggregate Mean Pooling**: Técnica de agregação usada para transformar um conjunto de vetores em um único vetor representativo via média:

$$v_{mean} = \left(\frac{1}{n}\right) \sum_{i=1}^{n} v_i$$

2.3 Re-ranqueadores

Re-ranqueadores são modelos que, dado um par (q, D), atribuem um *score* usado para refinar a ordenação dos documentos. Integram pipelines de recuperação em duas etapas, nas quais um modelo esparso ou denso recupera os *top-k* candidatos e o re-ranqueador reordena esse conjunto. Na prática, adotam predominantemente arquiteturas **cross-encoder**, que se distinguem das demais abordagens conforme ilustrado a seguir:

Bi-encoder
Codifica consulta e documento separadamente.
Menor custo computacional e menor precisão.
Indexação e processamento *offline*.
Similaridade via cosseno.

Cross-encoder
Processa consulta e documento em conjunto.
Maior custo computacional e maior precisão.
Modela interações *token* a *token*.
Similaridade aprendida $f_{\theta(q,d)} : (q, d) \mapsto \mathbb{R}$.

3 Trabalhos Relacionados

Este estudo se fundamenta em um conjunto amplo de contribuições da literatura de Recuperação da Informação (RI), abrangendo desde a construção de bases de dados nacionais até avanços em métodos lexicais, densos e híbridos.

- **JurisTCU** [1]: Conjunto de dados brasileiro para RI jurídica, contendo julgamentos anotados, metadados estruturados e diretrizes de indexação, servindo de referência para pesquisas na área de LIR.
- **BERT-based Dense Retrievers Require Interpolation with BM25** [2]: Demonstra que recuperadores densos dependem de interpolação com BM25 para desempenho consistente, ressaltando a complementaridade entre sinais lexicais e semânticos.
- **Análise da Eficácia de Fine-Tuning de Embeddings** [3]: Avaliação do impacto do fine-tuning em modelos densos (GTE, Jina, Gemma, Qwen) e técnicas como Matryoshka Learning, discutindo ganhos de desempenho e custo computacional.
- **When Documents Are Very Long, BM25 Fails** [4]: Introduz a variante BM25L e evidencia suas vantagens na recuperação de documentos extensos, validando diferenças por meio de testes estatísticos.
- **BMX – Entropy-weighted Similarity and Semantic-enhanced Lexical Search** [5]: Propõe o BMX, que combina pesos lexicais ajustados por entropia com query augmentation, ampliando a efetividade da busca lexical.
- **Ulysses Tesemô** [6]: Dataset legislativo da Câmara dos Deputados, com aproximadamente 105 mil documentos e 692 consultas anotadas, adequado para estudos de recuperação e análise de consultas.
- **An Analysis of Fusion Functions for Hybrid Retrieval** [7]: Estudo sobre funções de fusão em pipelines híbridos, incluindo TM2C2 e comparações com RRF.
- **Ranx** [8]: Biblioteca Python voltada para avaliação, comparação e fusão de rankings, com suporte a diversas métricas e algoritmos de combinação.
- **Pyserini** [9]: Toolkit para experimentação reprodutível em RI com métodos lexicais, densos e híbridos, incluindo pseudo-relevance feedback como RM3.

4 Proposta

Este trabalho propõe uma análise comparativa entre modelos esparsos, semânticos e híbridos no contexto jurídico brasileiro. A seguir são as abordagens avaliadas:

Modelos Esparsos

- BM25+ (*offset* positivo)
- BMX (*entropy-weighted*)
- Lucene (suavização do IDF)
- ATIRE (normalização alternativa)
- Pyserini com RM3 (PRF)
- BM25 Robertson (Baseline)

Modelos Semânticos

Modelos Base
- Qwen-Embedding-0.6B
- embeddinggemma-300m
- gte-multilingual-base
- jina-embeddings-v3

Variantes Pré-treinadas
- qwen-pgm-pairs
- gemma-pgm-pairs
- gte-pgm-pairs

Treinadas no *dataset* da PGM-Rio, conforme [3]

Para os modelos híbridos:

Modelos Híbridos

Algoritmos de Fusão		*Estratégias de Normalização*	*Reranqueadores*
• *Reciprocal-Rank Fusion* (RRF)		*Min-Max Scaling*	• bge-reranker-base
• *Multiple Number of Zeros* (MNZ)		*Max Normalization* (MNZ)	• gte-multilingual-reranker
• *Weighted* MNZ (WMNZ)	+	*Sum Normalization*	
• *Weighted* SUM (WSUM)		*Z-Score Mean-Unit Variance* (Z-MUV)	
• *Mixed* = WMNZ + WSUM		*Rank-Based Normalization*	
E muitos outros algoritmos de fusão ...		*E outras estratégias de normalização ...*	

4.1 Hipóteses

A análise experimental busca testar as seguintes hipóteses:

Hipótese 1. *Busca Híbrida × Baseline.*
Métodos de busca híbrida, que combinam BM25 e recuperadores densos por meio de algoritmos de fusão ou re-ranqueadores, resultam em melhorias estatisticamente significativas em relação ao BM25 *baseline* e ao melhor recuperador denso isolado.

Hipótese 2. *Métricas rasas × Métricas profundas.*
Segundo o artigo base [2], os ganhos relativos obtidos pela busca híbrida são mais expressivos nas métricas profundas (R@1000, nDCG@1000), em comparação às métricas rasas.

5 Implementação

5.1 Base de Dados

O estudo emprega o **JurisTCU**, *dataset* de *Legal Information Retrieval* composto por 16.045 decisões do TCU e 150 consultas anotadas com julgamento de relevância. As consultas compreendem três grupos: (i) Consultas reais submetidas por usuários do sistema, (ii) consultas sintéticas em formato de palavras-chave e (iii) consultas sintéticas em formato de pergunta, ambas geradas por LLMs. Neste trabalho, todas as consultas foram avaliadas conjuntamente para produzir uma estimativa global de desempenho.

5.2 BM25

A *pipeline* de pré-processamento implementada consiste nas seguintes etapas:

1. Persistência dos campos de indexação do artigo no banco de dados;
2. Extração de arquivos HTML usando `BeautifulSoup4`;
3. Normalização de termos jurídicos específicos (ex. “art. nº” → “artigo número”);
4. Aplicação do RSLP Stemmer desenvolvido para língua portuguesa;

As implementações utilizam diferentes bibliotecas especializadas: `bm25s` (biblioteca moderna e performática para BM25), `pyserini` (para BM25 com RM3) e `baguetter` do MixedBread (para BMX).

5.3 Sentence Embeddings

O processamento dos *embeddings* segue o fluxo abaixo:

1. Segmentação dos documentos em passagens por meio da biblioteca `langchain-text-splitters` com `chunk_size` = 1024;
2. Computação dos *embeddings* com dimensão de 768 para cada segmento usando os modelos selecionados;
3. Agregação dos *embeddings* usando *Aggregate Mean Pooling*;

Os *embeddings* são armazenados no DuckDB, um banco de dados embarcado OLAP otimizado para operações analíticas.

5.4 Reranqueadores

O processo de re-ranqueamento consiste em:

1. União e deduplicação dos resultados das etapas anteriores (BM25 e *embeddings*);
2. Re-ranqueamento dos resultados desconsiderando os *scores* anteriormente atribuídos, computando novos *scores* com modelos como `gte-multilingual-reranker`.

Os re-ranqueadores processam conjuntamente a consulta a cada documento candidato, produzindo novos *scores* que refletem a relevância semântica do par.

5.5 Estudo de Ablação

Esta etapa consiste na parametrização sistemática dos modelos avaliados. Para os métodos esparsos, o foco recai para os fatores de suavização do BM25, tais como o k_1 que controla a saturação da frequência de termos e b que regula a normalização pelo comprimento do documento. No caso dos métodos híbridos, a ablação incide sobre os algoritmos de fusão disponibilizados pela biblioteca `ranx`, os pesos empregados na combinação entre buscadores e as respectivas estratégias de normalização.

Visto isso, adotaram-se três etapas complementares:

1. *Random Search* para BM25: Realizou-se uma busca estocástica nos hiperparâmetros $k_1 \in [0.3, 3.0]$ e $b \in [0.0, 1.0]$, avaliando-se cada variante do BM25 e seus parâmetros específicos (b, k_1, δ, α, β). Retive-se exclusivamente a configuração com maior MAP.
2. Seleção dos candidatos: Concluída a busca dos hiperparâmetros do BM25, selecionou-se o modelo de *embeddings* com maior MAP no conjunto integral de consultas. Definiram-se, assim, os dois sistemas submetidos às fases subsequentes de re-ranqueamento e fusão.
3. *Grid Search* para fusão híbrida: Executou-se combinação simétrica entre algoritmos de fusão (RRF, MNZ, WMNZ, WSUM, ...) e estratégias de normalização (`min-max`, `sum`, `zmuv`, `rank`, `borda` ...) totalizando 132 configurações avaliadas.

A seguir, apresenta-se um esquema das etapas realizadas.

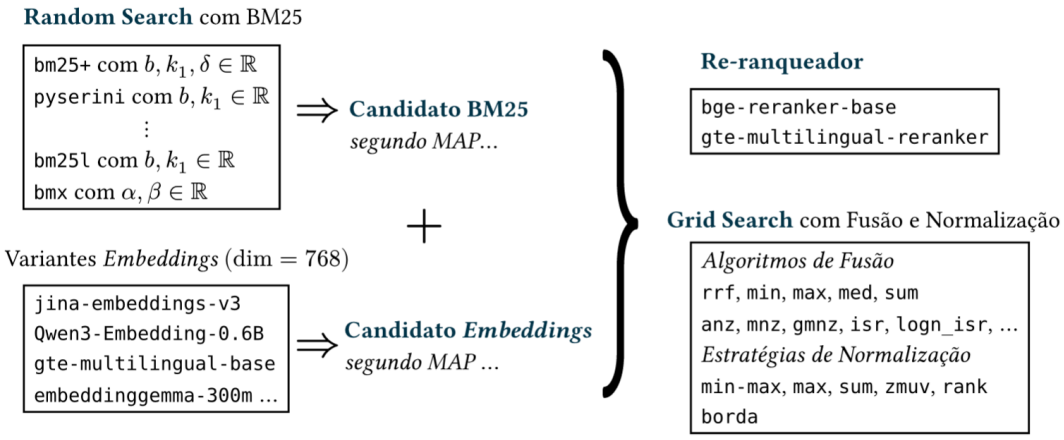

Diagrama 1. 1ª Fase: *Busca de hiperparâmetros e seleção de candidatos.*
2ª Fase: *Re-ranqueamento com resultado dos candidatos ou fusão.*

6 Resultados

Os resultados são apresentados em dois grupos de métricas, em linha com [2]:

- **Métricas rasas**: Avaliam a qualidade do topo do *ranking* ($k \leq 10$), incluindo P@K, R@K, F1@10, Hit@10 e nDCG@K;
- **Métricas profundas**: Consideram o *ranking* em maior profundidade, como MAP, R-Prec, R@100/ R@1000 e nDCG@100/nDCG@1000;

Observação: Por simplicidade, não foi possível disponibilizar todas as métricas calculadas no artigo presente. Portanto, refira-se ao repositório.

Neste trabalho, o MAP é adotado como métrica-síntese global, enquanto P@10, R@10 e nDCG@10 caracterizam a qualidade no topo da lista e R@1000/nDCG@1000 indicam a capacidade de cobertura em profundidade. A tabela a seguir resume os resultados para todos os modelos avaliados:

Tabela 1. MAP, métricas rasas e profundas para os grupos 1–3. Valores em **negrito** indicam o melhor resultado dentro de cada categoria (Esparso, Semântico, Re-ranqueadores, Fusão). Valores em dourado indicam a melhor métrica global em cada coluna.

Tipo	Modelo	MAP	Métricas Rasas ($k \leq 10$)			Métricas Profundas	
			P@10	R@10	nDCG@10	R@1000	nDCG@1000
Esparso	BM25+ (k_1=1,8, b=0,75, δ=1,5)	**0.309**	**0.341**	**0.292**	**0.440**	0.871	0.638
	BM25L (k_1=1,0, b=0,75, δ=1,5)	**0.309**	0.336	0.288	0.438	**0.875**	**0.639**
	Lucene (k_1=2,5, b=0,6)	**0.309**	0.336	0.288	0.438	**0.875**	**0.639**
	ATIRE (k_1=2,5, b=0,6)	0.308	0.336	0.288	0.438	**0.875**	**0.639**
	BMX (k_1=2.5, b=0,7, α=1,0, β=0.0)	0.305	0.335	0.286	0.434	0.872	0.636
	Pyserini (k_1=2,0, b=0,5)	0.302	0.339	0.289	0.439	0.864	0.636
	Robertson (Baseline) (k_1=2,5, b=0,7)	0.296	0.325	0.279	0.417	0.869	0.621
Semântico	jina-embeddings-v3	**0.402**	**0.443**	**0.376**	**0.532**	**0.908**	**0.702**
	Qwen-Embedding-0.6B	0.373	0.402	0.340	0.511	0.877	0.684
	gte-multilingual-base (Alibaba)	0.329	0.372	0.313	0.458	0.870	0.644
	gte-lamdec-pairs	0.300	0.337	0.284	0.416	0.863	0.607
	gemma-lamdec-pairs	0.154	0.193	0.162	0.240	0.714	0.412
	qwen-lamdec-pairs	0.143	0.167	0.141	0.216	0.689	0.393
Re-ranqueadores	gte-multilingual-reranker-base	**0.364**	**0.401**	**0.339**	**0.504**	0.919	**0.697**
	bge-reranker-base	0.195	0.219	0.183	0.279	0.883	0.513
Fusão	*Mixed* (WMNZ + WSUM) (norm=sum)	0.462	0.493	0.420	0.605	0.919	0.764
	Sum Fusion (norm=sum)	0.461	0.492	0.419	0.604	0.919	0.764
	Weighted Sum (norm=sum)	0.461	0.492	0.419	0.604	0.919	0.764
	MNZ (Min-Non-Zero) (norm=sum)	0.461	0.492	0.419	0.604	0.919	0.763
	GMNZ (Geometric MNZ) (norm=sum)	0.461	0.492	0.419	0.604	0.919	0.763

Observação: A métrica P@1000 foi omitida pois assumiu o valor 0,011 para praticamente todos os modelos e, desse modo, não agregou informações relevantes às análises.

A partir da Tabela 1, verifica-se que os algoritmos de fusão obtiveram os melhores resultados globais. O *Mixed* (WMNZ + WSUM) com normalização `sum` alcançou MAP de 0,462 superando todas as demais abordagens em métricas rasas e profundas. Como esperado, os modelos semânticos superaram os lexicais puros. O `jina-embeddings-v3` apresentou o maior MAP na categoria semântica $(0,402)$ e dominou tanto as métricas rasas quanto as profundas quando comparado às variante do BM25, constituindo um recuperador denso altamente competitivo.

Esse resultado é consistente com o artigo original do JurisTCU [1], no qual os *embeddings* são computados apenas sobre o campo de resumo (`summary`), enquanto o BM25 opera sobre o texto integral (`enunciado`). Tal configuração favorece modelos semânticos ao reduzir o ruído lexical. Por outro lado, o desempenho superior do Jina sobre o GTE diverge dos achados de Vargas [3], que reportou nDCG@20 de 0,533 para o GTE contra 0,517 para o Jina em similaridade fraca. As diferenças podem ser atribuídas às diferenças entre os *datasets* e o objetivo das consultas.

Notavelmente, os re-ranqueadores apresentaram desempenho inferior aos melhores modelos semânticos, em particular, o `gte-multilingual-reranker-base` obteve MAP de 0,364, abaixo do `jina-embeddings-v3` (0,402). Esse achado sugere que a estratégia de unir e deduplicar os resultados dos sistemas léxicos e semânticos descrita na Seção 5.4, descartando os *scores* originais, não foi eficaz.

As variantes pré-treinadas no *dataset* jurídico da PGM-Rio com sufixo `-lamdec-pairs` não apresentaram o desempenho esperado. Os modelos `gemma-lamdec-pairs` e `qwen-lamdec-pairs` obtiveram MAP de 0,154 e 0,143, respectivamente. Tais resultados são inferiores até mesmo à *baseline* BM25 Robertson (0,296). Isso sugere que o *fine-tuning* em um domínio jurídico específico pode não generalizar para outros.

Entre os modelos esparsos, as variantes do BM25 apresentaram desempenho similar entre si, com MAP variando de 0,296 (Robertson) a 0,309 (BM25+, BM25L, *Lucene*), indicando baixa sensibilidade à escolha da variante após a etapa de otimização dos hiperparâmetros.

Por fim, os ganhos da fusão híbrida foram mais expressivos nas métricas rasas ($+36,7\%$) do que nas profundas ($+16,27\%$), conforme a Tabela 2. Essa diferença de 20,5 pontos percentuais indica que a combinação de sinais léxicos e semânticos beneficia principalmente a qualidade dos primeiros resultados apresentados ao usuário.

Tabela 2. Comparação de ganhos: Métricas rasas × profundas

Categoria	**Ganho Médio**
Métricas Rasas	+36.7%
Métricas Profundas	+16.2%
Diferença	+20.5pp

Observação: O ganho foi calculado com base em todas as métricas rasas obtidas no código original e, portanto, não se restringe às métricas expostas na Tabela 1.

6.1 Hiperparâmetros

A Figura 2 ilustra a evolução do MAP ao longo das 570 configurações avaliadas. A variação é expressiva: de 0,008 (RRF com normalização `borda`) a 0,453 (*Mixed* com normalização `sum`), evidenciando alta sensibilidade à escolha do algoritmo de fusão e da estratégia de normalização.

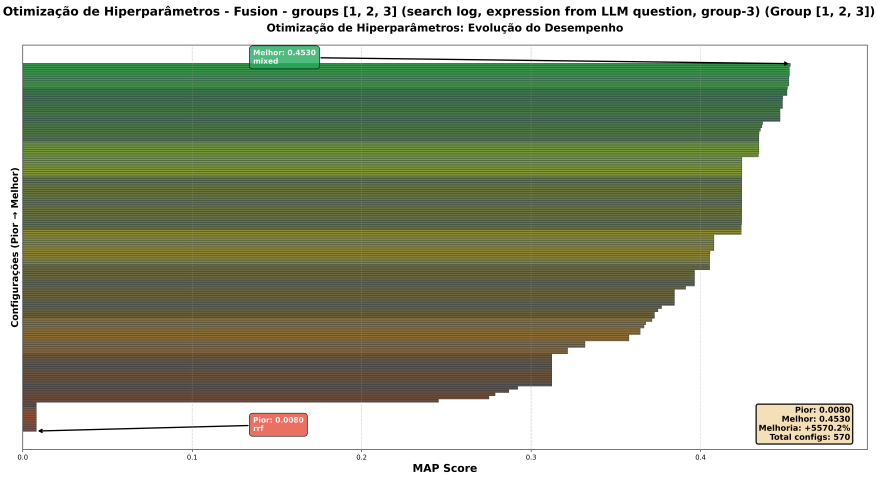

Figura 2. *Comparação entre estratégias de fusão e normalização segundo o* MAP *durante a etapa de Grid Search.*

6.2 Desempenho

A Tabela 3 apresenta o tempo de execução por consulta para cada modelo. A ordenação dos resultados corresponde às expectativas: Modelos esparsos são os mais rápidos (mediana $\sim 1,2$ ms), seguidos dos semânticos ($\sim$ 83-107 ms), algoritmos de fusão ($\sim$ 129-138 ms) e, por fim, re-ranqueadores ($\sim 6.424 - 82.165$ ms).

Comparando as duas abordagens híbridas, os algoritmos de fusão são significativamente mais eficientes que os re-ranqueadores. O *Mixed* (WMNZ + WSUM) apresenta mediana de $136,6$ ms, enquanto o `bge-reranker-base` requer 6.424 ms, ou seja, cerca de 47 *vezes* mais lento.

Tabela 3. Tabela. Tempo de execução de consultas para o grupo 1, 2, 3. Valores em **negrito** indicam o menor tempo (mais rápido) dentro de cada categoria. Valores em verde indicam o menor tempo global.

Tipo	Modelo	Tempo de Consulta (ms)				
		Média	Mediana	P95	P99	Máx
Esparso	BM25+	1.248	1.219	1.429	1.517	4.071
	BM25L	1.274	1.249	1.463	1.803	4.062
	Lucene	1.265	1.231	1.500	1.636	4.037
	ATIRE	1.302	1.231	1.696	2.440	4.284
	BMX	3.920	1.734	2.231	2.930	325.695
	Pyserini BM25 com RM3	41.844	41.084	48.478	52.060	124.087
	BM25 Robertson (Baseline)	1.351	1.293	1.823	2.966	3.890
Semântico	jina-embeddings-v3	95.789	93.657	100.710	106.097	371.476
	Qwen-Embedding-0.6B	102.256	101.723	107.587	111.873	165.264
	gte-multilingual-base	**83.274**	**82.622**	**90.318**	**92.333**	129.927
	gte-lamdec-pairs	91.982	91.799	99.843	102.641	**107.524**
	gemma-lamdec-pairs	105.013	104.405	111.565	114.790	120.042
	qwen-lamdec-pairs	107.881	106.342	119.141	125.193	126.472
Re-ranqueadores	gte-multilingual-reranker-base	85311.906	82165.312	111014.787	132483.799	148444.723
	bge-reranker-base	**6433.176**	**6424.173**	**6847.512**	**6985.753**	**7190.475**
Fusão	*Mixed* (WMNZ + WSUM)	137.362	136.604	143.119	148.792	194.907
	Sum Fusion	131.342	130.926	137.001	142.557	200.150
	Weighted Sum	138.660	137.988	145.639	148.696	211.934
	MNZ (Min-Non-Zero)	**129.172**	**128.746**	**134.893**	**137.543**	**187.551**
	GMNZ (Geometric MNZ)	134.204	133.607	140.578	145.428	208.638

6.3 Configuração

Os experimentos foram executados no ambiente descrito na Tabela 4. Os testes e avaliações utilizaram Python 3.13 e CUDA 13.0, e os modelos densos e re-ranqueadores foram servidos por meio do framework PyTorch 2.9.1.

Tabela 4. *Configurações do Ambiente Computacional utilizado.*

Componente	Configuração
CPU	Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz
RAM	94GB
GPU	2 × NVIDIA RTX (8 GB cada)
Disco	907GB (/)
SO	Debian GNU/Linux 12 (bookworm)

6.4 Comparação

A Tabela 5 reproduz os resultados do artigo JurisTCU apenas para o Grupo 3 (consultas em formato de pergunta). Este cenário favorece métodos léxicos, visto que as perguntas tendem a compartilhar vocabulário com os documentos, elevando a *baseline* do BM25.

Tabela 5. Comparação com o artigo JurisTCU (Grupo 3). Valores em dourado indicam o melhor resultado de cada métrica; Trechos em azul correspondem aos modelos deste trabalho.

Tipo	Modelo	Métricas @10			
		P@10	R@10	MRR	nDCG@10
Esparso	BM25 (baseline)	0.388	0.345	0.918	0.533
	BM25.dT5q	0.408	0.362	0.939	0.556
	BM25.Syn(GPT3.5)	0.406	0.361	0.915	0.546
	BM25.Syn(GPT4o)	0.408	0.363	0.934	0.552
	BM25.Syn(Llama3)	0.396	0.352	0.923	0.541
	BM25.dT5q.Syn(GPT35)	0.416	0.369	0.919	0.557
	BM25.dT5q.Syn(GPT4o)	0.416	0.369	0.940	0.564
	BM25.dT5q.Syn(Llama3)	**0.420**	**0.372**	0.929	**0.564**
	BM25+ (k1=3.0, b=0.6)	0.402	0.358	**0.943**	0.551
Semântico	BERT.pt.TCU	0.202	0.180	0.608	0.288
	BERT.pt.large	0.222	0.196	0.607	0.289
	BERT.pt.large.legal	0.348	0.307	0.868	0.466
	BERT.ml	0.344	0.305	0.792	0.452
	OpenAI.small	0.482	0.425	0.917	0.609
	OpenAI.large	0.472	0.415	0.915	0.608
	Qwen-Embedding-0.6B	**0.500**	**0.440**	**0.982**	**0.654**
Fusão	Fusão MNZ (norm=zmuv, k=20)	0.554	0.490	0.990	0.698

No bloco semântico, o modelo `Qwen-Embedding-0.6B`, menor e aberto, supera os *embeddings* da OpenAI utilizados no artigo, em todas as métricas. Assim, como o Qwen, mesmo sem ajuste adicional, já supera todos os modelos semânticos do artigo original, a comparação direta entre a fusão proposta e o melhor modelo do estudo de referência torna-se assimétrica. Apesar disso, os resultados da busca híbrida demonstram ganhos reais superando o Qwen em P@10 (+10, 8%) e nDCG@10 (+6,7%).

No bloco esparso, o BM25+ com $k_1 = 3,0$ e $b = 0,6$, parâmetros obtidos via *Random Search*, atinge MRR de 0,943, superando todas as variantes do artigo nessa métrica. O resultado dá indícios de que a otimização de hiperparâmetros pode alcançar desempenho comparável às técnicas de expansão de consulta (dT5q, Syn) sem a complexidade adicional dessas abordagens.

6.5 Teste de Hipótese

Para verificação dos resultados, foi conduzido o teste não paramétrico de Wilcoxon pareado, que dispensa pressupostos de normalidade na distribuição dos dados, comparando a mediana da diferença entre as precisões médias das consultas. Adotou-se um nível de significância de $\alpha = 0,05$.

Tabela 6. Resultados dos Testes de Wilcoxon ($\alpha = 0.05$)

Comparação	Estatística W	P-valor
BM25 Robertson × Fusion Mixed	165.00	< 0.000001
Embedding Jina × Fusion Mixed	3104.00	< 0.000002

Os resultados do teste de Wilcoxon ($\alpha = 0,05$) rejeitam as hipóteses nulas de equivalência entre:

- BM25 Robertson (*baseline*) e *Fusion Mixed* ($p < 0.000001$);

- `jina-embeddings-v3` e *Fusion Mixed* ($p < 0.000002$).

Portanto, conclui-se que o método de fusão supera significativamente tanto a *baseline* esparsa quanto o melhor modelo denso isolado.

7 Conclusão

Este trabalho apresentou uma análise comparativa de métodos de recuperação da informação no contexto jurídico brasileiro, avaliando abordagens lexicais, semânticas e híbridas sobre o *dataset* JurisTCU.

Os resultados confirmam a Hipótese 1: a busca híbrida supera consistentemente métodos puramente lexicais ou semânticos. O algoritmo *Mixed* (WMNZ + WSUM) alcançou MAP de 0,462, representando ganhos de 56% sobre a *baseline* Robertson (0,296) e 15% sobre o melhor modelo denso isolado, o `jina-embeddings-v3` (0,402). A significância estatística dessas diferenças foi confirmada pelo teste de hipótese de Wilcoxon.

Com relação à Hipótese 2, observou-se um comportamento distinto, os ganhos da fusão foram mais expressivos nas métricas rasas ($+36,7\%$) do que nas profundas ($+16,2\%$). Isso sugere que a combinação de sinais lexicais e semânticos beneficia principalmente os primeiros resultados.

Adicionalmente, constatou-se que: (i) modelos pré-treinados em domínios jurídicos específicos nem sempre generalizam para outros *datasets* da mesma área; (ii) a otimização de hiperparâmetros do BM25 pode alcançar desempenho comparável a técnicas de expansão de consulta; e (iii) algoritmos de fusão podem oferecer melhor eficiência para níveis superiores ou comparáveis de performance em comparação aos re-ranqueadores.

8 Trabalhos Futuros

Como extensão deste estudo, identifiquei as seguintes direções de pesquisa:

- *Retrieval-Augmented Knowledge Graph*: Expansão de consulta baseada em grafo de conhecimento gerado com entidades nomeadas extraídas via GliNER. Permite enriquecer a recuperação com relações estruturadas entre conceitos e termos do *corpus*.
- Validação em outros *datasets*: Replicação no *dataset Ulysses Relevance Feedback Corpus* (105.669 documentos), para avaliar a generalização das conclusões para outros domínios jurídicos.

References

1. Fernandes, L.C., Santos Ribeiro, L. dos, Castro, M.V.B. de, Silva Pacheco, L.A. da, Oliveira Sandes, E.F. de: JurisTCU: A Brazilian Portuguese Information Retrieval Dataset with Query Relevance Judgments. arXiv preprint. (2025).
2. Wang, S., Zhuang, S., Zuccon, G.: BERT-based Dense Retrievers Require Interpolation with BM25 for Effective Passage Retrieval. In: Proceedings of the 2021 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR~'21). pp. 317–324. Association for Computing Machinery, Virtual Event, Canada (2021). https://doi.org/10.1145/3471158.3472233.
3. —: Análise da Eficácia de Fine-Tuning de Embeddings no Contexto Jurídico Brasileiro. —. (202 AD).
4. Lv, Y., Zhai, C.X.: When Documents Are Very Long, BM25 Fails!. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11). pp. 1103–1104 (2011). https://doi.org/10.1145/2009916.2010070.
5. Li, X., Lipp, J., Shakir, A., Huang, R., Li, J.: BMX: Entropy-weighted Similarity and Semantic-enhanced Lexical Search. arXiv preprint. (2024).
6. —: Building a Relevance Feedback Corpus for Legal Information Retrieval in the Real-Case Scenario of the Brazilian Chamber of Deputies. In: — (202 AD).
7. Bruch, S., Gai, S., Ingber, A.: An Analysis of Fusion Functions for Hybrid Retrieval. arXiv preprint. (2022).
8. Bassani, E.: ranx: A Blazing-Fast Python Library for Ranking Evaluation and Comparison. In: ECIR (2). pp. 259–264. Springer (2022). https://doi.org/10.1007/978-3-030-99739-7_30.
9. Lin, J., Ma, X., Lin, S.-C., Yang, J.-H., Pradeep, R., Nogueira, R.: Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In: Proceedings of the 44th Annual

International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021). pp. 2356–2362 (2021).