Bilingual Hybrid Judicial-Opinion Retrieval for Puerto Rico and Texas: Dense, Lexical, and LLM-Augmented

Andrés Pérez Cabezas
Master in Computer Science
Advisor: Lisabel Rodríguez Espinosa, Ph.D.
Polytechnic University of Puerto Rico
Graduate Project EXPO May 2026

Abstract *— This article presents the design, implementation, and evaluation of a bilingual English/Spanish information retrieval prototype over judicial opinions from Puerto Rico and Texas courts. Judicial opinions are ingested from the CourtListener REST API, cleaned, segmented into paragraph-aware chunks, embedded with a multilingual sentence-transformer model, and indexed in Facebook AI Similarity Search (FAISS) for dense retrieval, with a parallel Best Matching 25 (BM25) index supporting lexical retrieval. A weighted Reciprocal Rank Fusion (RRF) layer combines semantic and lexical rankings. An optional layer adds query rewriting and cross-encoder reranking using a locally hosted Gemma 3 large language model. On a stratified held-out split of thirty-two queries, weighted hybrid RRF reaches a normalized discounted cumulative gain at rank 10 (nDCG@10) of 0.5615, while cross-encoder reranking raises this value to 0.7680 at roughly twenty times higher latency. The system preserves end-to-end traceability to source judicial opinions and is presented as a research prototype, not a legal-advice product.*

Keywords *— Bilingual Information Retrieval, Dense Embeddings, Hybrid Search, Reciprocal Rank Fusion.*

INTRODUCTION

Legal research over judicial opinions is a fundamental task for attorneys, clerks, and academic researchers, but the tools available for genuinely bilingual jurisdictions remain limited. Puerto Rico courts publish judicial opinions predominantly in Spanish, while Texas courts publish judicial opinions almost exclusively in English. A researcher comparing judicial opinions across these jurisdictions must either issue parallel queries against jurisdiction-specific systems or accept the limitations of monolingual ranking on a mixed-language corpus. Even when legal search platforms provide keyword or semantic search, they often do not expose enough of the underlying retrieval mechanism, intermediate scores, or evaluation protocol to characterize where and why a search engine fails on a given query.

Since this prototype was initially developed and evaluated, CourtListener has introduced public semantic search for case law on its website [1]. This development further confirms the practical relevance of meaning-based legal retrieval. The present work remains distinct in scope: it focuses on a reproducible bilingual retrieval prototype for Puerto Rico and Texas judicial opinions that compares dense, lexical, hybrid, and optional LLM-augmented ranking methods under a fixed evaluation protocol.

This project addresses the transparency and bilingual-evaluation gap by building a bilingual retrieval prototype over a curated corpus of judicial opinions from Puerto Rico and Texas courts sourced from the CourtListener REST API [2]. The system combines three complementary retrievers: dense semantic search using multilingual sentence embeddings [3], lexical search using BM25 [4], and a weighted Reciprocal Rank Fusion (RRF) [5] layer that blends both rankings. An optional layer adds query rewriting and cross-encoder reranking [6] using the Gemma 3 large language model [7] hosted locally on Ollama. The contributions of the work are threefold: (i) a reproducible end-to-end pipeline for bilingual judicial-opinion retrieval; (ii) a held-out evaluation comparing dense, lexical, hybrid, and LLM-augmented retrieval configurations on one

hundred curated queries with human-reviewed relevance judgments; and (iii) an empirical observation that cross-encoder reranking provides a substantial ranking-quality lift but at a latency cost that makes it unsuitable as the default interactive mode on the tested local workstation.

Prior work in legal information retrieval has explored both sparse and dense methods independently but rarely in bilingual settings. The TREC Legal Track [8] established standard evaluation paradigms for English-language legal retrieval, while recent neural approaches such as ColBERT [9] and DPR [10] have demonstrated that dense retrieval can outperform BM25 on general-domain benchmarks. In the multilingual space, mBERT [11] and its distilled variants have enabled cross-lingual transfer for retrieval tasks, but evaluations have focused predominantly on European language pairs or newswire corpora rather than legal text. Bilingual legal retrieval poses distinctive challenges: legal terminology is domain-specific and jurisdiction-dependent, judicial opinions mix procedural boilerplate with substantive reasoning, and the same legal concept may be expressed in structurally different ways across languages. The present system draws on these foundations and contributes a reproducible research prototype that integrates dense embeddings, BM25, Reciprocal Rank Fusion, and optional LLM reranking over a mixed English-Spanish corpus of Puerto Rico and Texas judicial opinions.

The remainder of this article describes the corpus and pipeline, the system architecture, the evaluation methodology, the observed results, and the limitations and future-work implications of the prototype.

CORPUS AND DATA PIPELINE

The corpus is sourced from CourtListener, a free legal database operated by the Free Law Project [2]. Ten court sources were selected to provide bilingual and cross-jurisdictional coverage while supporting comparison between Puerto Rico and Texas judicial opinions. The selected sources include the Supreme Court of Puerto Rico, the Puerto Rico Court of Appeals, the U.S. District Court for the District of Puerto Rico, the Supreme Court of Texas, the Texas Court of Criminal Appeals, the Texas Courts of Appeals, and the four federal judicial districts of Texas. These sources are used as legal research materials for information-retrieval experimentation, not as a statement that all included judicial opinions carry the same precedential authority. In particular, the U.S. District Court for the District of Puerto Rico and the federal district courts of Texas are trial-level federal courts whose decisions may be useful for retrieval and comparison, but they are not equivalent in precedential weight to appellate or supreme court opinions.

Ingestion is throttled by the client at one request per second with exponential back-off on transient failures, within CourtListener API usage constraints [2], and capped at two thousand judicial opinions per court source to constrain disk usage and ingest time. The resulting processed corpus comprises 20,128 cleaned judicial opinions covering the years 1990 through 2026, with the noted exception of the Puerto Rico Court of Appeals (1995-2004) and the Supreme Court of Puerto Rico (1990-2015), where the per-court cap binds before reaching the present day.

The language distribution of the corpus reflects the jurisdictional split. Judicial opinions from the three Puerto Rico-related sources account for approximately six thousand opinions and are predominantly in Spanish, with occasional bilingual or English-language opinions from the U.S. District Court for the District of Puerto Rico. The seven Texas-related sources contribute the remaining approximately fourteen thousand opinions, nearly all in English. Opinion lengths vary substantially across courts, with appellate opinions tending to be longer than federal district court opinions and orders, which often include procedural rulings. The average chunk count per opinion is approximately seventeen, with the variance in opinion length directly affecting how each opinion is segmented at the chunking stage.

Because this system is a research prototype rather than a legal-authority classifier, it does not currently rank retrieved judicial opinions by

precedential weight, binding authority, or court hierarchy. Retrieved results may include binding appellate opinions, persuasive decisions, or trial-level federal opinions depending on the selected court source. Future versions of the corpus should more explicitly distinguish local Puerto Rico courts, Puerto Rico-related federal courts, Texas state courts, and Texas federal courts. They should also prioritize appellate sources such as the U.S. Court of Appeals for the First Circuit for Puerto Rico-related federal appellate authority and add metadata fields for court level, jurisdiction, language, and precedential status.

The cleaning stage strips HTML markup, repairs common Unicode mojibake (garbled characters produced when text is decoded with the wrong encoding, e.g., "café" rendered as "café") using the ftfy library ("fixes text for you," a Python utility that detects and reverses common mis-decoded byte sequences in Unicode strings), removes PDF extraction artifacts, and normalizes whitespace. Judicial opinions are then split into overlapping paragraph-aware chunks with a target length of 512 tokens and an overlap of 64 tokens, with a minimum chunk length of 50 tokens to avoid pathologically short matches. The chunked corpus contains 351,564 text segments. Each chunk is embedded with the paraphrase-multilingual-MiniLM-L12-v2 Sentence-Transformers model [12], [3], a distilled multilingual variant of BERT [13], producing 384-dimensional vectors that are L2-normalized and indexed in FAISS [14] using IndexFlatIP, which under unit-norm vectors is mathematically equivalent to exact cosine similarity. A parallel BM25 index [4] is constructed in memory using the rank-bm25 library, sharing the same chunk-level metadata schema.

SYSTEM ARCHITECTURE

The system is organized as a layered pipeline. The retrieval layer fetches ranked candidate chunks from the indexed corpus using semantic, lexical, or hybrid scoring. An optional LLM layer wraps the retrieval baseline with a query rewriter and a cross-encoder reranker that can be enabled independently. A FastAPI service and a Next.js user interface expose the resulting configurations through a uniform result schema. The three subsections that follow describe each layer in turn.

RETRIEVAL LAYER

The retrieval layer is the component of the pipeline that takes a user query and returns a ranked list of candidate chunks; it sits between the indexed corpus and the optional rerank and LLM stages. It exposes three primary modes. Semantic retrieval issues the query embedding to FAISS and returns the top-k chunks ranked by cosine similarity, applying optional court and date filters. BM25 retrieval scores all chunks using a whitespace tokenization scheme. Hybrid retrieval performs a weighted Reciprocal Rank Fusion of both result lists [5]. For a candidate judicial opinion d, let rs denote its rank position in the semantic result list and rb its rank position in the BM25 result list. The fused score is then given by:

$$score(d) = \frac{ws}{(k + rs)} + \frac{wb}{(k + rb)} \quad (1)$$

Here d denotes a single candidate judicial opinion being scored that is, one specific document drawn from the indexed corpus whose final ranking position is being computed. Each candidate d receives one fused score, and the result list is built by evaluating this expression over every opinion that appeared in either source list. The term score(d) denotes the fused Reciprocal Rank Fusion score assigned to candidate judicial opinion d, which is used to order the result list in descending order. The weights ws and wb control the relative contribution of the semantic and BM25 retrievers respectively and satisfy ws + wb = 1; the constant k is the standard RRF dampening term, which limits the influence of any single high-ranked result by adding a fixed offset to each rank position before inversion. The retriever ranks rs and rb are one-indexed positions within their respective result lists. In this work, ws = 0.7, wb = 0.3, and k = 30; these values were selected on a development split. Before fusion, each source list is deduplicated by opinion identifier in first-seen rank order that is, if multiple chunks from the same

judicial opinion appear in a result list, only the highest-ranked chunk is kept and the rest are discarded. This deduplication step is required to maintain reproducibility against the standalone reference implementation; without it, repeated chunks from the same judicial opinion can inflate the fused score and shift later candidates. This issue produced a measurable evaluation drift before being identified and pinned by a real-data parametrized regression test.

Figure 1 shows the implementation pattern corresponding to (1). Before Reciprocal Rank Fusion is applied, each retriever output is deduplicated at the opinion level so repeated chunks from the same judicial opinion do not inflate the fused score. The final score is then computed as a weighted sum of semantic and BM25 reciprocal-rank contributions.

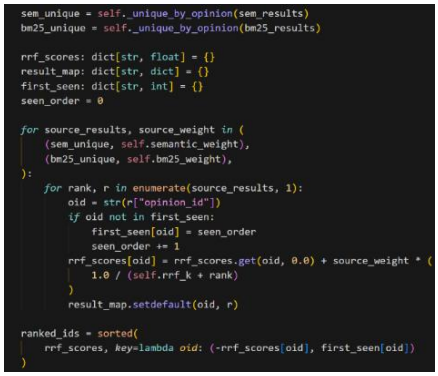

Figure 1
Weighted Hybrid Retrieval with Opinion-Level Deduplication

Beyond the core ranking, the retrieval layer supports optional metadata filters for court identifier and date range. Court filters are applied as post-retrieval masks on the FAISS result set and as pre-retrieval index partitions on BM25, reflecting the different access patterns of each backend. Date filters restrict results to judicial opinions filed within a user-specified window and are implemented symmetrically across both retrievers. When both filters are active, a conjunctive AND is applied. The filtering mechanism is intentionally simple: it applies hard masks rather than soft boosts, so filtered results are strictly excluded rather than demoted. This design choice prioritizes transparency, since users can be certain that filtered-out opinions do not influence the ranking, at the cost of forgoing potential relevance signals from partially matching metadata. Each result returned by the system carries the full provenance chain: the originating opinion identifier, the court, the filing date, the chunk index within the opinion, and the raw similarity or BM25 score, enabling downstream consumers to audit the ranking at the chunk level.

Optional LLM Layer

Two optional stages wrap the hybrid baseline. The query rewriter expands the user query into a richer bilingual form, including English and Spanish synonyms and common legal terminology, by issuing a prompt to a locally hosted Gemma 3 12B model [7] via the Ollama runtime. The rewritten query then drives standard hybrid retrieval. The cross-encoder reranker fetches a deeper hybrid pool, with a default pool_k of 50, and scores each query-chunk pair on a 0 to 10 scale using the same model, sorts the candidates by descending score, and returns the top-k [6]. Both stages cache prompt outputs by SHA-256 of the full prompt to make repeated runs deterministic and inexpensive. On provider failure, the rewriter returns the original query untouched, and the reranker returns the first top-k baseline candidates with llm_score = 0, so plain hybrid behavior is preserved when the LLM daemon is unavailable.

API and User Interface

A FastAPI service exposes /health, /courts, /search, and /opinion/{opinion_id}/chunks endpoints. The /search endpoint accepts six retriever values: semantic, bm25, hybrid, hybrid+rewrite, hybrid+rerank, and hybrid+rewrite+rerank. It returns a uniform result schema augmented with two optional fields: a top-level rewritten_query string and a per-result llm_score. The /opinion/{opinion_id}/chunks endpoint reads the per-opinion processed parquet, applies a render-only display-time cleanup pass that strips CourtListener pagination markers and rejoins overlapping chunks, and returns a single concatenated full_text together

with per-chunk byte offsets so a caller can highlight a specific chunk by string slice. The canonical user interface is a Next.js application that defaults to hybrid retrieval and exposes a seven-pill retriever selector: Hybrid, Semantic, BM25, a side-by-side Semantic plus BM25 mode, and the three LLM-augmented modes. This allows all retriever configurations evaluated in this work to be user-selectable for inspection. Selecting a result opens a side panel that renders the full opinion text with the matched chunk highlighted and a deep link back to the originating CourtListener opinion. A Streamlit application is retained as a development tool. All result cards likewise link directly back to the originating CourtListener opinion to preserve human-verifiable traceability.

Evaluation Methodology

The evaluation is designed to compare retriever configurations on a controlled bilingual query set under conditions that are reproducible from the project artifacts alone. The three subsections that follow describe, in order, how the queries and relevance labels were assembled, how the tune and held-out splits and the retrieval metrics are defined, and the reproducibility safeguards adopted after two issues were uncovered during evaluation.

Query Set and Relevance Judgments

The evaluation set comprises one hundred queries, balanced fifty/fifty across jurisdictions (Puerto Rico vs. Texas) and languages (English vs. Spanish). Queries were drafted to span common legal concepts in family, civil, administrative, criminal, and constitutional law. For each query, relevant opinion identifiers were collected manually by inspecting CourtListener results and verifying topical relevance against the opinion text. The annotation effort was performed by a single annotator; inter-annotator agreement was therefore not measured. Because the evaluation is intended to compare retrieval configurations within a research prototype rather than establish a definitive legal benchmark, the single-annotator relevance set is treated as a practical experimental control rather than a final gold standard.

Splits and Metrics

A stratified tune/held-out split is fixed in splits.json with a seed of forty-two, yielding thirty-two held-out queries, with eight queries per jurisdiction-language stratum. This split is treated as a fixed scientific artifact and is not regenerated during evaluation. All hyperparameter selection, including RRF weights, k, and pool sizes, is performed on the tune split; final reported numbers come exclusively from the held-out split. Metrics are computed at the opinion level, with chunk-level results deduplicated, and include precision at rank 10 (P@10), recall at rank 10 (R@10), mean reciprocal rank (MRR), and normalized discounted cumulative gain at rank 10 (nDCG@10).

The choice of metrics reflects different facets of retrieval quality relevant to legal research. Precision at rank 10 measures the proportion of returned results that are relevant, corresponding to the practical question of how much irrelevant material a researcher must sift through. Recall at rank 10 measures the proportion of all relevant judicial opinions that appear in the top ten, capturing the system's ability to surface the full set of pertinent judicial opinions rather than only a few exemplars. Mean reciprocal rank rewards systems that place the first relevant result near the top of the list, which matters for researchers who scan results sequentially and stop early. Normalized discounted cumulative gain at rank 10 assigns graded credit by rank position, penalizing relevant results that appear lower; it is the primary metric reported because it balances precision and recall in a single number and is standard in the information retrieval literature. Latency is reported as a wall-clock average to characterize the practical cost of each retriever configuration, since interactive legal research tools are expected to respond within a few seconds.

Reproducibility Considerations

Two notable bugs were uncovered during evaluation and are documented as part of the project

record. First, the hybrid retriever did not deduplicate input chunks before fusion, which caused a measurable nDCG@10 drift between the standalone weighted-RRF reference script and a refactored composed retriever. Second, an evaluation script read query and split files using the Python default codepage, cp1252 on Windows, silently corrupting accented Spanish characters and altering the resulting embeddings. Both issues were fixed and are now pinned by real-data parametrized tests.

Figure 2 shows the explicit UTF-8 decoding used when loading evaluation queries and split files. This fix prevents accented Spanish terms from being silently corrupted by the Windows default codepage, which previously affected embeddings and retrieval scores.

```
queries_data = json.loads(Path("src/eval/queries.json").read_text(encoding="utf-8"))
queries = queries_data["queries"]

splits = json.loads(splits_path.read_text(encoding="utf-8"))
```

Figure 2
Explicit UTF-8 Decoding for Evaluation Files

RESULTS

Held-out results on the thirty-two-query split are summarized in Table 1. Weighted hybrid RRF improves over BM25 alone on every metric and over semantic retrieval alone on precision, recall, and nDCG@10, with a near-tie on MRR. On this held-out split, cross-encoder reranking provides the strongest observed ranking quality, raising nDCG@10 by 0.207 absolute over the weighted hybrid baseline. The bilingual query rewriter, by contrast, underperforms the baseline on this benchmark, and combining rewriting with reranking yields a result inferior to reranking alone.

Table 1
Held-out Retrieval Performance (n = 32 queries)

	P@10	R@10	MRR	nDCG@10	Latency
Weighted Hybrid RRF	0.4969	0.2630	0.7808	0.5615	~2.5 s
Hybrid + rerank	0.7031	0.4108	0.9010	0.7680	~52 s
Hybrid + rewrite	0.3969	0.1967	0.6615	0.4496	~8 s
Hybrid + rewrite + rerank	0.4469	0.2517	0.7781	0.5068	~55 s

The latency measurements were collected on the author's local workstation running Windows 64-bit with an AMD Ryzen 9 7950X3D 16-Core Processor, 64 GB of RAM, and an NVIDIA GeForce RTX 4090 with 24 GB of VRAM, using Ollama with the gemma3:12b checkpoint in FP16. Latency values reported here are local wall-clock averages on this workstation, not a generalized benchmark. The cross-encoder cost dominates because reranking is implemented as fifty independent prompt calls per query, one per pool candidate. Listwise reranking, which scores K candidates in a single prompt, is identified as a high-priority optimization but has not been validated in this work.

The query-rewriter regression deserves comment. Inspection of rewritten queries suggests that aggressive synonym expansion shifts the BM25 ranking away from the expected term-overlap region, while semantic embeddings of the longer rewritten query drift toward generic legal vocabulary rather than the specific factual matter at issue. A more conservative rewriter, one that emits synonyms only for short queries or augments rather than replaces the original query, is a plausible remedy but is left for future work.

For context, the standalone retrievers that feed the hybrid layer were also evaluated independently on the held-out split. Semantic-only retrieval achieved a P@10 of 0.4813, an R@10 of 0.2460, an MRR of 0.7840, and an nDCG@10 of 0.5492. BM25-only retrieval achieved a P@10 of 0.3812, an R@10 of 0.2414, an MRR of 0.5836, and an nDCG@10 of 0.4177. These numbers suggest that semantic retrieval performs better than BM25 on this corpus, likely because the multilingual embedding model captures cross-lingual similarity that whitespace-tokenized BM25 cannot. They also show that weighted hybrid fusion improves over either standalone retriever on precision, recall, and nDCG@10, with MRR effectively tied with semantic retrieval.

A per-jurisdiction and per-language slice of the held-out results would clarify whether retrieval quality differs systematically between Texas and Puerto Rico queries, or between English and Spanish queries. The expectation is that the BM25 component benefits from exact English term overlap in Texas judicial opinions, while Spanish queries

against Puerto Rico judicial opinions rely more heavily on the semantic component to bridge vocabulary variation. Cross-encoder reranking may narrow some of these gaps by scoring query-passage pairs contextually, but a rigorous slice analysis is left for future work. With only eight queries per jurisdiction-language stratum on the held-out split, point estimates would carry wide confidence intervals.

DISCUSSION

The strongest practical observation from the evaluation is that, on this corpus and query distribution, weighted hybrid RRF performs reliably on this benchmark and requires no machine-learning infrastructure beyond the embedding model and the BM25 index. It is therefore the operational default for the prototype. Cross-encoder reranking yields markedly better rankings on the held-out split, but a per-query latency of roughly fifty-two seconds places it out of reach for interactive demonstrations on commodity hardware. Bilingual query rewriting, by contrast, is the only configuration that performed worse than the weighted hybrid baseline, a useful negative result that motivates more careful prompt design before any future use.

The held-out split contains thirty-two queries, which is small. Bootstrap confidence intervals around small-sample nDCG estimates are wide, and the magnitude of the rerank lift should be interpreted accordingly. The split is bilingual and balanced by jurisdiction, however, which provides some assurance that the result is not driven entirely by one language or one court family.

A second source of caution is the relevance-judgment process. Oracle identifiers were collected by a single annotator examining CourtListener results, and so the evaluation measures retrieval performance against a reasonable but not adversarial gold set. Larger, multi-annotator relevance pools would produce a more defensible benchmark and would be a prerequisite for any claim that the system is fit for use beyond demonstration.

The prototype is not directly comparable to commercial legal-research platforms, which are tuned on private data and proprietary editorial signals not available to academic projects. From the perspective of an external researcher, however, the present system exposes every retrieval signal, including cosine similarity scores, BM25 scores, RRF fusion weights, and optional LLM reranking scores, in a uniform JSON schema end to end. This supports inspection, diagnosis of ranking failures at the chunk level, and modification by external researchers. Commercial platforms typically do not expose comparable diagnostics to outside researchers, though direct comparison is outside the scope of this work.

From a deployment perspective, the system's hardware requirements are modest for the baseline configuration. The FAISS index for 351,564 vectors at 384 dimensions consumes approximately 540 megabytes of memory, and, measured in the author's local development environment, the BM25 index used approximately 1.2 gigabytes. Both fit on a machine with 64 gigabytes of RAM in the author's workstation. In local timing tests, CPU query embedding completed in under 50 milliseconds. The LLM stages, however, require a GPU with at least 12 gigabytes of VRAM for the Gemma 3 12B model in FP16, and the per-query reranking latency of fifty-two seconds makes the cross-encoder impractical as a synchronous endpoint. A more mature deployment would likely serve the hybrid baseline as the default interactive mode and offer reranking as a slower batch-style option for researchers willing to wait for higher-quality results.

Finally, the system is explicitly framed throughout as a research prototype that surfaces retrieved judicial opinions for human review. It is not, and is not represented as, a legal-advice product.

LIMITATIONS

Several limitations bound the scope of the present work. First, the per-court cap of two thousand judicial opinions truncates Puerto Rico's coverage and creates a date asymmetry with the

Texas courts; lifting the cap is constrained by CourtListener rate limits and quotas. Second, the held-out split is small, and the relevance judgments were produced by a single annotator, so the reported metrics should be interpreted as prototype-level evidence rather than as a definitive retrieval benchmark. Third, BM25 tokenization is whitespace-based and does not perform accent folding or Spanish stemming, so lexical matching across diacritic variation depends heavily on the embedding model. Fourth, API filter validation is permissive, and malformed dates or unknown court identifiers may produce empty result sets without descriptive error messages. Fifth, the LLM stages depend on a local Ollama daemon and a specific model checkpoint; reproducibility across machines requires both. Finally, the system has not been hardened for production use: there is no authentication, no rate limiting, no usage-telemetry pipeline, and no deployment recipe beyond the local development workflow.

CONCLUSION

This article has described a transparent bilingual judicial-opinion retrieval prototype that combines dense and lexical signals through weighted Reciprocal Rank Fusion, optionally augmented by LLM-based query rewriting and cross-encoder reranking. On a small but balanced bilingual benchmark, weighted hybrid RRF is the best-performing configuration that meets interactive latency expectations, and it is therefore the operational default for the prototype. Cross-encoder reranking sets the highest observed ranking quality among the tested configurations, though this work does not claim it is an absolute upper bound. Bilingual query rewriting, by contrast, is a cautionary example of how naive prompt engineering can degrade rather than improve retrieval. The system preserves end-to-end traceability from the user's query to the originating CourtListener judicial opinion and is offered as a research artifact for further academic study rather than as a substitute for professional legal research.

FUTURE WORK

Future work will pursue four directions. First, scaling the corpus beyond the per-court cap and rebalancing Puerto Rico's date coverage to reach the present day would allow a more rigorous evaluation of generalization. Second, listwise cross-encoder reranking, which scores multiple candidates in a single prompt, is a promising strategy for reducing reranking latency, but tune-split validation should precede any held-out claim. Third, a more conservative query-rewriting strategy, including length-conditional triggering and additive rather than replacement-based rewrites, is required before LLM rewriting can be considered a positive contribution. Finally, a larger and multi-annotator relevance pool, combined with a user study involving practicing attorneys or law-school researchers, would convert the present technical evaluation into a stronger assessment of research utility.

Beyond these four primary directions, several secondary improvements are under consideration. Accent folding and Spanish stemming in the BM25 tokenizer would address a known weakness in lexical matching across diacritic variation; the Snowball stemmer for Spanish is a natural candidate. Incorporating citation-graph features, such as the number of times a judicial opinion has been cited by subsequent decisions, as a static quality signal could improve ranking without increasing query-time latency, since citation counts can be precomputed during ingestion. On the interface side, adding a citation-network visualization that shows how retrieved judicial opinions relate to each other through judicial citations would provide researchers with a structural understanding that a flat ranked list cannot convey. Finally, containerizing the full stack with Docker Compose and publishing a reproducible deployment recipe would lower the barrier to replication and allow other researchers to extend or adapt the system for additional jurisdictions and languages.

REFERENCES

[1] R. Gao. (2026, May 4). *Semantic Search Is Now on CourtListener* [Online]. Available: https://free.law/2026/05/04/semantic-search-on-courtlistener/. [Accessed: May 11, 2026].

[2] Free Law Project. (2026). *REST API, v4.4, CourtListener Documentation* [Online]. Available: https://www.courtlistener.com/help/api/rest/. [Accessed: May 8, 2026].

[3] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. on Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, Nov. 2019, pp. 3982-3992. [Online]. Available: https://aclanthology.org/D19-1410/.

[4] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," in *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333-389, 2009. [Online]. Available: https://www.staff.city.ac.uk/~sbrp622/papers/foundations_bm25_review.pdf.

[5] G. V. Cormack et al., "Reciprocal rank fusion outperforms Condorcet and individual rank learning methods," in *Proc. 32nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Boston, MA, USA, Jul. 2009, pp. 758-759. [Online]. Available: https://dl.acm.org/doi/10.1145/1571941.1572114.

[6] R. Nogueira and K. Cho, "Passage re-ranking with BERT," arXiv preprint, arXiv:1901.04085, Jan. 2019. [Online]. Available: https://arxiv.org/abs/1901.04085.

[7] Gemma Team, Google DeepMind, "Gemma 3 Technical Report," arXiv preprint, arXiv:2503.19786, Mar. 2025. [Online]. Available: https://arxiv.org/abs/2503.19786.

[8] J. R. Baron et al., "TREC-2006 Legal Track overview," in *Proc. Fifteenth Text REtrieval Conf. (TREC 2006)*, 2006. [Online]. Available: https://trec.nist.gov/pubs/trec15/papers/LEGAL06.OVERVIEW.pdf.

[9] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT," in *Proc. 43rd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2020, pp. 39-48. [Online]. Available: https://dl.acm.org/doi/10.1145/3397271.3401075.

[10] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in *Proc. 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769-6781. [Online]. Available: https://aclanthology.org/2020.emnlp-main.550/.

[11] T. Pires et al., "How multilingual is multilingual BERT?" in *Proc. 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, Jul. 2019, pp. 4996-5001. [Online]. Available: https://aclanthology.org/P19-1493/.

[12] Sentence Transformers. (2019, Aug. 27). *Paraphrase-multilingual-MiniLM-L12-v2* [Online]. Available: https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2. [Accessed: May 8, 2026].

[13] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, USA, Jun. 2019, pp. 4171-4186. [Online]. Available: https://aclanthology.org/N19-1423/.

[14] J. Johnson et al., "Billion-scale similarity search with GPUs," in *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535-547, Jul.-Sep. 2021. [Online]. Available: https://arxiv.org/pdf/1702.08734.