PowerLLM: A Local-First Retrieval-Augmented System for Legal Document Analysis

1. Introduction

1.1 Motivation

Legal professionals routinely work with large volumes of complex documents, including contracts, statutes, and case materials. A significant portion of this work involves locating relevant clauses, interpreting legal language, and cross-referencing provisions across documents, which is both time-consuming and error-prone.

Traditional keyword-based search tools provide limited support in this context. Legal queries often involve nuanced phrasing, implicit references, or structural elements such as sections and clauses that are not easily captured by simple lexical matching. As a result, those tasked with document review must frequently perform manual review, navigating long documents to identify the exact span of relevant information.

Recent advances in retrieval-augmented generation (RAG) systems have enabled more effective document-level question answering by combining language models with retrieval mechanisms. However, most existing legal AI tools are cloud-based, proprietary, and designed for enterprise use, raising concerns regarding data privacy, cost, and accessibility. This creates a gap for practitioners who require secure, local processing of sensitive documents.

These constraints pose a dilemma for junior practitioners, who handle the bulk of document-intensive tasks but often lack access to expensive, enterprise-grade legal AI. This accessibility gap may inadvertently drive them toward publicly available AI tools (ChatGPT websites etc.), potentially leading to the unauthorised transmission of confidential client data and breaching professional standards of confidentiality.

These limitations highlight the need for a system that operates fully offline while maintaining acceptable retrieval accuracy and latency, supporting data privacy requirements and addressing practical constraints where access to secure and cost-effective tools is limited.

1.2 Project Aims

This project sets out to address these challenges by building PowerLLM, a local-first, LLM-powered document retrieval and question-answering system designed with the needs of junior lawyers in mind. Rather than requiring users to skim through entire documents, PowerLLM allows practitioners to pose natural language questions, such as "What are the termination conditions in this contract?" or "Which clause governs liability?" and receive concise, grounded answers drawn directly from the uploaded documents.

2. Background & Related Work

2.1 Commercial Legal AI Tools

Harvey AI is a specialised generative AI platform that leverages large language models (LLMs) fine-tuned for legal reasoning and document analysis. By integrating with enterprise-grade models, Harvey processes millions of queries to assist in contract drafting and litigation support. Despite its efficacy in reducing manual labor, Harvey operates as a closed-source, cloud-based service, which raises concerns regarding data sovereignty and subscription costs for smaller legal practices.

Similarly, Thomson Reuters' CoCounsel (formerly Casetext) and Lexis+ AI represent the integration of Retrieval-Augmented Generation (RAG) with proprietary legal databases. Unlike general-purpose AI, these tools ground their outputs in verified case law and statutes to minimise hallucinations. Although these platforms allow analysis of user uploaded documents, they remain tightly bound to vendor controlled ecosystems and require sensitive data to be sent to external servers. This reliance makes them unsuitable for highly confidential materials that must be processed in fully air gapped or locally controlled environments.

3. Implementation

This section describes the key design components of the system, focusing on processing, retrieval, and generation stages.

3.1 Pre Process

PDF/DOCX Files Converter

The pre-processing pipeline converts raw documents (PDF and DOCX) into structured Markdown representations using Docling. This step standardises document formats and preserves structural information such as sections and layout, which improves downstream chunking and retrieval quality.

For scanned or image-based documents, OCR is automatically applied to recover textual content before further processing.

To reduce noise in embedding and retrieval, a lightweight watermark filtering mechanism is applied to remove non-informative visual elements extracted during document parsing. This helps prevent irrelevant patterns from affecting semantic indexing.

Watermark Detection and Remover

Legal documents often contain visible or semi transparent watermarks indicating confidentiality or ownership, which can interfere with readability and automated processing. When processed with Docling, these elements may be extracted as images, introducing noise that degrades

embedding quality and retrieval accuracy. Proper handling of such documents is therefore necessary to preserve textual integrity and ensure reliable performance.

For each extracted image associated with a processed document, watermark detection is performed using a lightweight heuristic pipeline. Candidate images are first screened by a size-based rule, where images below a predefined width or height threshold are classified as watermarks. For reference-based detection, ORB keypoints and binary descriptors are extracted from the candidate image and matched against descriptors precomputed from reference watermark images using a Brute-Force matcher with *Hamming distance*. Matches with distance below a fixed threshold are retained, and the candidate image is classified as a watermark when the number of retained matches exceeds a predefined threshold. Detected watermark images are then excluded from subsequent OCR and indexing stages.

3.2 Models

Local Models

The system prioritises the use of locally deployable language models that can run efficiently on consumer-grade laptops while still delivering acceptable performance for legal retrieval and question-answering tasks. Model selection is therefore guided by a balance between **computational cost** and **practical utility**.

Comparison of Memory Usage of Models

Model Size	**FP16 Memory (GPU/ Unified)**	**8-bit Quantised**	**4-bit Quantised**	**Typical Laptop Feasibility**
0.6B	~1.2–1.5 GB	~0.6–0.8 GB	~0.3–0.5 GB	Very easy (CPU/GPU)
4B	~8–10 GB	~4–5 GB	~2–3 GB	Feasible (16GB RAM)
9B	~18–22 GB	~9–11 GB	~4.5–6 GB	Borderline (needs optimisation)

Model memory consumption scales approximately linearly with parameter size, making full-precision deployment impractical for larger models on consumer hardware. Quantisation techniques, particularly 4-bit representations, significantly reduce memory requirements while maintaining acceptable performance for downstream tasks. As a result, models in the 4B range represent a practical balance between capability and deployability on devices with 16 GB of RAM, while smaller models (e.g., 0.6B) are suitable for lightweight tasks such as embedding and query processing.

Supported Models

Including language models directly within the application package is impractical for real-world deployment due to their large size, platform-specific dependencies, and the difficulty of maintaining and updating them alongside the application. Instead, the system adopts a decoupled design in which models are hosted externally and accessed through a unified, OpenAI-compatible interface.

Users configure models by specifying parameters such as `base_url`, `model_name`, and optional credentials, allowing the system to interact seamlessly with both local inference servers (e.g., LM Studio, oMLX) and remote APIs. This approach improves flexibility, enabling users to select models that match their hardware constraints and privacy requirements. Still, the system is optimised for Qwen-3.5-4B-MLX as chat models and Qwen-3-Embedding-0.6B as embedding models. Prompt templates are specifically tuned for these models to improve instruction adherence and reduce overly brief or incomplete responses.

3.3 Retrieval Methods

Dense Retrieval (Similarity Retrieval)

Dense retrieval, also referred to as similarity retrieval, retrieves document chunks based on semantic similarity in the embedding space (Karpukhin et al., 2020). The user query and document chunks are encoded into dense vector representations using the selected embedding model, and retrieval is performed by measuring vector similarity between the query embedding and indexed chunk embeddings. This allows the retrieval component to identify relevant legal content even when the query and the supporting text are expressed using different wording.

Query Builder for BM25

Before BM25 retrieval, the input query is reformulated into a retrieval-oriented lexical query by a query builder. The query is first normalised through basic cleaning and lowercasing. A language model then extracts a small set of concise keywords or short phrases that are likely to appear in the source documents, while retaining exact names, dates, numbers, citations, and legal references when present. In addition, legal references are expanded into alternative surface forms, such as converting Arabic numerals into Chinese numeral expressions. The final BM25 query is constructed by combining the normalised original query, the extracted keywords, and the expanded legal-reference variants, which improves lexical matching for legal document retrieval.

BM25 Retrieval

BM25 serves as a lexical retrieval baseline that quantifies the relevance between queries and document chunks through exact term matching (Robertson & Zaragoza, 2009). The ranking mechanism incorporates term frequency (TF) and inverse document frequency (IDF), while applying document length normalisation to mitigate bias. This approach is advantageous in

scenarios requiring high lexical fidelity, such as retrieving statutory terminology, specific clause identifiers, or domain-specific nomenclature.

Reciprocal Rank Fusion (RRF)

Within the retrieval pipeline, Reciprocal Rank Fusion (RRF) is used to combine results from dense and BM25 retrieval (Cormack et al., 2009), a rank-based aggregation method that is robust to differences in score scale across retrieval components. For a candidate chunk d, the fused score is computed as:

$$\mathrm{RRF}(d) = \sum_{r \in R} \frac{1}{k + \mathrm{rank}_r(d)}$$

Where R denotes the set of retrieval result lists, $\mathrm{rank}_r(d)$ is the rank position of chunk d in retriever r, and k is a smoothing constant that reduces the dominance of top-ranked items while still rewarding consistently well-ranked candidates. In the retrieval setting, R here contains two ranked lists: one from dense vector similarity retrieval and one from BM25 retrieval.

3.4 Generation

Retrieval-Augmented Generation

The system adopts a Retrieval-Augmented Generation (RAG) framework (Lewis et al., 2020) to support document-oriented question answering over user-provided legal documents. In this setting, responses must be grounded in document-specific content rather than relying on the model's internal parametric knowledge.

Retrieved document chunks are concatenated and provided as context to the language model, which is instructed to generate answers strictly based on the retrieved evidence. This design constrains generation to external context, reducing hallucination and improving traceability.

System prompts are available in Appendix.2.

Source Citation

The retrieval documents and the bounding boxes of original texts in PDF files will be sent to frontend at the end of answering. By clicking the citations, users are allowed to view the cited contents with highlight in original documents and verify.

4. Evaluation

This evaluation focuses exclusively on the retrieval pipeline. Components outside retrieval, including document pre-processing, response generation, and end-to-end system behaviour, are outside the scope of this study and were not evaluated.

4.1 Datasets

To provide a reliable evaluation of the retrieval pipeline, this project utilises the LegalBench-RAG as evaluation datasets. LegalBench-RAG is a variant of the original LegalBench datasets, with format optimisation for RAG pipelines.

This dataset contains 6,858 query-answer pairs extracted from a corpus over 79 million characters. Due to computational limitations, we randomly sample 200 questions from each data group to create evaluation subsets. The architecture maps queries from the foundational LegalBench project back to their original locations within four distinct legal sub-domains:

- **CUAD**: Focuses on contract clause extraction.
- **MAUD**: Addresses complex mergers and acquisitions documentation.
- **PrivacyQA**: Concerns the interpretation of organisational privacy policies.
- **ContractNLI**: Evaluates natural language inference within contractual frameworks.

Although these datasets capture the primary task settings the system is designed to support, real-world documents encountered in practice may vary significantly in format, structure, and expression. As such, the system's performance may differ when applied to unconstrained user-provided inputs. Further limitations are discussed in the **Section 7.** Limitation section.

4.2 Metrics

The evaluation considers four aspects: retrieval effectiveness, ranking quality, evidence coverage, and efficiency.

- **Recall@k (Recall@20, Recall@final_k)**: Measures the proportion of relevant documents retrieved within the top-k results.
- **Hit@k (Hit@1, @3, @5):** Indicates whether at least one relevant document appears within the top-k results, reflecting early retrieval usefulness.
- **Precision@k (Precision@5, @10, @20)**: Measures the proportion of relevant documents among the retrieved results, capturing noise in the retrieved set.
- **MRR (Mean Reciprocal Rank)**: Evaluates ranking quality by measuring how early the first relevant document appears.
- **Fully Covered / Partial+**: Fully Covered / Partial+: Final-window coverage metrics that measure whether the retrieved final context fully covers all gold answer spans, or at least partially covers one or more of them.
- **Overlap (Mean_overlap_ratio_at_final_k_mean)**: Measures partial coverage by computing the maximum character-level overlap between retrieved chunks and gold answer spans, averaged across questions.

4.3 Results

Baseline (BM25 only approach)

```
"retrieval_params": {
    "chunking_config": legal,
    "rewrite_query": true,
    "bm25_k": 20,
    "final_k": 10, # Keep the first 10
  },
```

Where "legal" refers to the legal-structure-aware chunking configuration described in Section 3. The baseline uses the same chunking configuration as the proposed system to isolate the effect of the retrieval strategy.

Benchmark using the Baseline Configuration

Metrics	**CUAD**	**PrivacyQA**	**ContractNLI**	**MAUD**	**Overall**
Recall@10	0.4118	0.6421	0.9717	0.0925	0.5295
Hit@1	0.1050	0.2371	0.3950	0.0150	0.1880
Hit@3	0.2400	0.4845	0.7150	0.0550	0.3736
Hit@5	0.3100	0.6340	0.8600	0.0850	0.4723
Precision@10	0.0220	0.0565	0.0388	0.0206	0.0345
MRR	0.1847	0.3658	0.5517	0.0473	0.2874
Fully Covered	0.3450	0.5103	0.9550	0.0600	0.4676
Partial+	0.4950	0.7938	0.9900	0.1450	0.6059
Not Covered	0.5050	0.2062	0.0100	0.8550	0.3940
Overlap	0.4257	0.7373	0.9751	0.1603	0.5746

Optimal Configuration:

```
"retrieval_params": {
    "chunking_config": legal,
    "rewrite_query": true,
    "similarity_k": 9,
    "bm25_k": 39,
    "final_k": 10,
    "rrf_k": 60
  },
```

This configuration is selected based on the tuning results presented in **Section 5**. Config Tuning.

Benchmark using the Optimal Configuration

Metrics	CUAD	PrivacyQA	ContractNLI	MAUD	Overall
Recall@10	0.6245	0.6877	0.9775	0.2295	0.6298
Hit@1	0.1650	0.3144	0.4150	0.0450	0.2349
Hit@3	0.4000	0.4948	0.8050	0.1400	0.4599
Hit@5	0.5500	0.6598	0.9350	0.2050	0.5874
Precision@10	0.0380	0.0664	0.0410	0.0357	0.0453
MRR	0.2849	0.3935	0.5819	0.1137	0.3435
Fully Covered	0.5300	0.5825	0.9650	0.1700	0.5619
Partial+	0.7150	0.8299	0.9900	0.3100	0.7112
Not Covered	0.2850	0.1701	0.0100	0.6900	0.2888
Overlap	0.6373	0.8084	0.9797	0.3422	0.6919

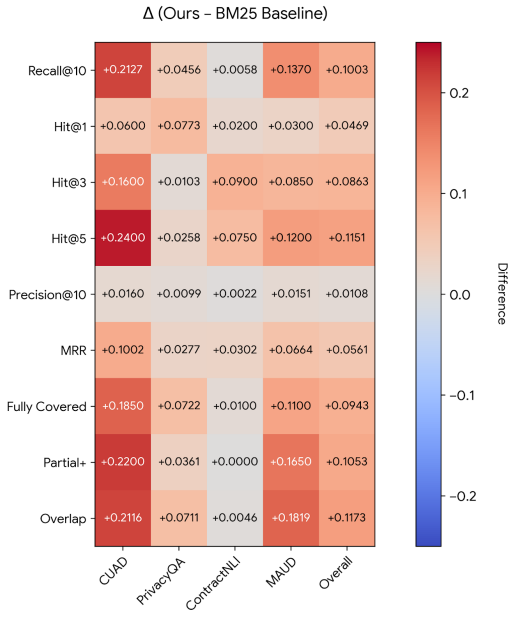


Figure 1. Retrieval performance comparison across datasets.

4.4 Discussion

The table presents a comparative quantitative analysis between a hybrid-retrieval methodology ("Ours") and a standard lexical baseline (BM25) across four domain-specific datasets (CUAD,

PrivacyQA, ContractNLI, MAUD) alongside an aggregate measure ("Overall"). The hybrid retrieval approach demonstrates improved retrieval effectiveness and context coverage across the aggregate metrics.

Dataset Variance

While the overall trend indicates substantial improvement over BM25, the magnitude of these gains varies substantially depending on the target corpus:

Significant Advancements (CUAD & MAUD): The model exhibits the highest performance delta on the CUAD and MAUD datasets. On CUAD, top-k retrieval metrics show substantial increases (e.g., Hit@5 improves by 0.2400), accompanied by consistent improvements in span overlap (Partial+ increases by 0.2200). MAUD similarly benefits from improvements in Overlap (+0.1819) and Recall@10 (+0.1370), although overall performance remains comparatively low. Such challenges are likely attributable to the multifaceted structure of MAUD documents, where salient information is often embedded within protracted sections and expressed through technical nomenclature, hindering the efficacy of precise retrieval mechanisms.

Moderate to Marginal Gains (PrivacyQA & ContractNLI): Improvements on PrivacyQA are consistent but more constrained, generally falling below a 0.0800 increase across most metrics. Conversely, performance gains on ContractNLI are marginal. This is likely because BM25 is inherently well-suited to this dataset, where queries often rely on exact terminology, clause-level references, and explicit lexical matches. As a result, the baseline already achieves near-saturated performance across several metrics (e.g., Recall@10 and Partial+), leaving limited room for improvement. Under such conditions, the additional semantic signals introduced by dense retrieval contribute only minor gains.

These variations suggest that retrieval performance is strongly influenced by document structure and task characteristics, rather than solely by retrieval strategy.

In summary, the retrieval results suggest that the hybrid retrieval pipeline is better suited to tasks such as contract information extraction and queries involving specific legal terms, which are aligned with the intended use cases of PowerLLM.

5. Config Tuning

5.1 Settings

The config tuning process uses the same metrics as in benchmark. The hyperparameter optimisation is performed using Optuna, a Bayesian optimisation framework. The search explores combinations of retrieval parameters by iteratively sampling configurations and maximising the defined quality score objective.

Due to computational constraints, the optimisation is conducted over a restricted number of trials on a subset of the CUAD dataset, with each trial evaluated independently using the same retrieval metrics as the main benchmark.

Quality Score

The goal for config tuning is to optimise the overall coverage while maintaining smaller chunking size and average context length.

The score is calculated using the following formula.

$$\begin{aligned}\text{quality\_score} = {} & 0.34 \times \text{recall\_at\_final\_k} \\ & + 0.22 \times \text{partial\_or\_better\_rate} \\ & + 0.18 \times \text{fully\_covered\_at\_final\_k} \\ & + 0.10 \times \text{mrr\_mean} \\ & + 0.08 \times \text{hit\_at\_5} \\ & + 0.08 \times \text{mean\_overlap\_ratio\_at\_final\_k}\end{aligned}$$

The main goal is to maximise the coverage(`partial-or-better-rate`), which measures the proportion of questions for which at least some relevant content was retrieved. This is the broadest coverage signal. It indicates whether the retrieval system found any useful information, regardless of completeness. It carries the highest weight in the objective as total misses are the most damaging failure mode for a RAG system since the generator cannot recover from an empty or irrelevant context.

One known limitation is that coverage-based metrics can improve mechanically when the final retrieved context becomes less selective, even without a genuine improvement in retrieval quality. Further consideration is discussed alongside proposed mitigations in `retrieval_config_tuning.py`.

For more details, please refer to `config_tuning_README.md` in repository.

Results

The dataset for config tuning is a 200 questions random sample of CUAD dataset.

Metrics	**Baseline**	**Best in Test**
chunk_size	2000	2000
chunk_overlap	400	400
similarity_k	10	9
bm25_k	10	39
final_k	10	10
rrf_k	60	60
Recall@FinalK	0.6183	**0.6245**

Metrics	**Baseline**	**Best in Test**
Hit@5	0.5300	**0.5500**
Precision@5	**0.0490**	0.0380
Precision@10	0.0365	**0.0380**
MRR	0.2764	**0.2848**
Fully Covered@10	0.5300	0.5300
Mean Overlap Ratio@10	0.6327	**0.6373**
Partial or Better	0.7100	**0.715**

We conducted a grid search to analyse the interaction between $bm25\_k$ and $similarity\_k$. The results indicate a relatively flat performance landscape overall, with no clear monotonic trend along either dimension. However, a locally optimal region can be observed when $similarity\_k$ is around 9, where the ranking metric achieves its highest values.

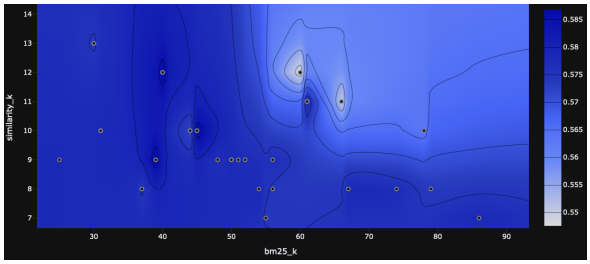


Figure 2. Contour map of retrieval performance across different values of k for BM25 and similarity scoring.

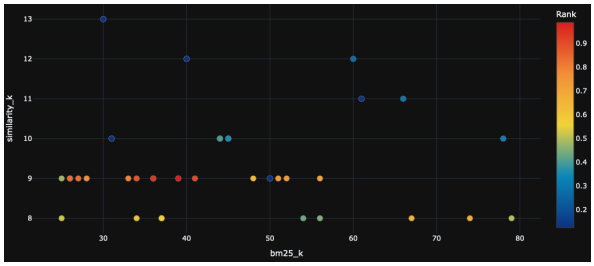


Figure 3. Comparative analysis of retrieval ranks under various hyperparameter combinations for BM25 and vector similarity. (*Note that some entries recorded as 0.0 are not actual zero scores, but were inadvertently introduced during the pruning process.*)

Larger values of $similarity\_k$ (e.g., above 10–11) are in several cases associated with weaker performance, although the effect is not strictly monotonic. This pattern suggests that expanding the dense candidate pool beyond a moderate size may introduce less relevant candidates, which

can negatively affect ranking quality. This aligns with the assumption that real-world legal RAG tasks frequently require precise terminology and that BM25 outperforms dense search in such situations.

Variations in $bm25\_k$ result in fluctuating performance without a consistent upward trend indicating diminishing returns from increasing the sparse retrieval depth. Overall, the results favour a conservative hybrid configuration where a relatively small dense candidate set is combined with a moderate BM25 pool.

Given the observed plateau in performance, further gains are unlikely to be achieved through expanding retrieval candidate sets alone. We hypothesise that tuning the relative weights of BM25 and dense retrieval in RRF may yield additional gains, although this remains to be validated in future experiments.

6. Demo Frontend App

For demonstration purposes, this project includes a lightweight frontend application built with the Electron framework. The frontend is designed to illustrate the interaction workflow of the system, rather than to provide a production-ready interface. As such, certain aspects such as scalability, robustness, and edge-case handling are not fully addressed.

6.1 Case Management

The system adopts a case-centric abstraction, where each case represents an isolated workspace containing its own document collection and retrieval context. This structure allows users to organise documents by legal matter and ensures that queries are scoped to the relevant case.

Document ingestion follows a simplified pipeline in which uploaded files are processed, converted into structured representations, and indexed for retrieval. The implementation focuses on demonstrating the core workflow rather than handling all failure modes or large-scale data management.

6.2 Conversational Query Interface

The AskAI module provides a thread-based conversational interface for interacting with the RAG pipeline, implemented using the Assistant UI framework to manage streaming message states and thread persistence. Conversations are organized as threads: each associated with a specific legal case that maintain complete message history, branch structures, and metadata in the browser's localStorage.

The interface supports streaming responses and maintains conversation history, enabling iterative querying over the same document set.

6.3 Source Citation

A citation highlighting mechanism seamlessly connects the conversational and document interfaces. This allows users to verify generated responses against their source material. When the RAG pipeline provides citations with bounding box coordinates the interface displays them as interactive links within the chat stream. Activating a citation opens the source PDF in a side panel navigates to the relevant page and overlays translucent highlights on the cited text.

7. Limitations

7.1 Possibly Infeasible Local Models

The central design principle in our design is local-first deployment and processing all data fully offline, while this offers meaningful advantages in terms of privacy and cost, it comes with an inherent trade-off in model capability.

From a capability perspective, LLMs benefit from scaling laws, where increased parameter count and training data lead to improved performance in reasoning, compositional generalisation, and instruction following. These properties are particularly critical in tasks involving multi-step inference, long-context understanding, and ambiguous query interpretation, which are common in legal and document-centric retrieval systems.

Empirical studies indicate that while small language models can achieve competitive performance on constrained or domain-specific tasks, larger models consistently demonstrate superior performance on complex, multi-step reasoning tasks, with improvements scaling with model size . In high-complexity domains such as legal text analysis, which involve long-context reasoning and specialised semantics, large models currently achieve stronger results on benchmark evaluations.

Specifically:

- They are less robust to long or noisy contexts, leading to weaker grounding in retrieval-augmented settings.
- They demonstrate limited ability in multi-hop reasoning and implicit relation extraction.
- They are more sensitive to prompt phrasing and distributional shifts.
- Their latency significantly increases with larger context windows.

These limitations mean that PowerLLM is best suited to retrieval-oriented tasks:

- Locate relevant clauses
- Surfacing specific dates or obligations
- Summarising targeted passages
- Parse content to structure formats

Rather than tasks requiring deep legal analysis or multi-document reasoning.

However, it is worth noting that despite these limitations, small language models have improved rapidly, with ongoing research continuing to enhance their performance and efficiency.

7.2 Hyperparameters

Optimisation Scope and Parameter Constraints

Although the retrieval configuration was tuned systematically using Optuna on the CUAD benchmark, the resulting parameter set should be regarded as a strong empirical approximation rather than a globally optimal solution.

To maintain computational feasibility under local-model constraints, the optimisation was performed over a deliberately restricted search space.

Specifically, high-dimensional variables such as `final_k` and the chunk-overlap ratio were held constant to prioritise system responsiveness. Since the RAG pipeline is designed for on-device execution, limiting `final_k` is essential to prevent excessive inference latency during the LLM generation phase, where a larger context window would disproportionately increase the computational overhead. Admittedly, this focus on operational efficiency means that superior parameter combinations may reside within the unexplored regions of the broader hyper-parameter landscape.

Potential Evaluation–Deployment Mismatch

Due to computational constraints, all evaluations were conducted on subsets of the full CUAD, ContractNLI, and PrivacyQA datasets, which may affect the generalisability of tuned configurations to the full corpus.

Another related limitation is that the retrieval config objective was retrieval-specific and evaluated on a CUAD subset of English commercial contracts. This means the selected parameters are biased toward that dataset and toward retrieval quality alone, and may not transfer equally well to other legal document types or to overall answer-generation performance. As such, the tuned configuration should be interpreted as a practical configuration for the evaluated setting, rather than a universally optimal choice.

8. Conclusion

In this work, we introduce PowerLLM, a local-first retrieval-augmented system designed to address the privacy and cost constraints of legal document analysis. By combining dense retrieval and BM25 through Reciprocal Rank Fusion, together with query rewriting and legal-specific preprocessing, the system achieves measurable gains in retrieval effectiveness and evidence coverage over lexical baselines, with the largest improvements observed in tasks involving complex document structures and domain-specific terminology.

Operating entirely on local infrastructure, however, introduces inherent trade-offs. The system's performance is constrained by the capacity of locally deployable models and remains

sensitive to task characteristics. In particular, the benefits of the hybrid retrieval pipeline are reduced in scenarios where lexical matching alone is sufficient.

Despite these limitations, the results indicate that a carefully designed, locally deployed retrieval pipeline can provide effective support for legal question answering while preserving data privacy. Looking ahead, the modular tuning framework offers opportunities to explore a broader configuration space, enabling more systematic optimisation and the development of adaptive retrieval strategies under improved computational settings.

References

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., *et al.* (2020). *Language models are few-shot learners.* Advances in Neural Information Processing Systems (NeurIPS).

Cormack, G. V., Clarke, C. L. A., & Büttcher, S. (2009). *Reciprocal rank fusion outperforms condorcet and individual rank learning methods.* Proceedings of SIGIR.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding.* Proceedings of NAACL-HLT.

Docling Project. (2024). *Docling: Document understanding toolkit.* Retrieved from https://github.com/docling-project/docling

Gao, L., Ma, X., Lin, J., & Callan, J. (2022). *Precise zero-shot dense retrieval without relevance labels.* Proceedings of ACL.

Guo, Y., Guo, M., Su, J., Yang, Z., Zhu, M., Li, H., Qiu, M., & Liu, S. S. (2024). *Bias in large language models: Origin, evaluation, and mitigation.* arXiv preprint arXiv:2411.10915.

Harvey AI. (2023). *Harvey AI legal platform.* Retrieved March 2026, from https://www.harvey.ai

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., *et al.* (2020). *Dense passage retrieval for open-domain question answering.* Proceedings of EMNLP.

LangChain. (2024). *LangChain documentation.* Retrieved from https://www.langchain.com

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., *et al.* (2020). *Retrieval-augmented generation for knowledge-intensive NLP tasks.* Advances in Neural Information Processing Systems (NeurIPS).

LexisNexis. (2023). *Lexis+ AI platform overview.* Retrieved March 2026, from https://www.lexisnexis.com/en-us/products/lexis-plus-ai.page

oMLX. (2024). *oMLX: High-performance local LLM inference framework.* Retrieved March 2026, from https://omlx.ai

Robertson, S., & Zaragoza, H. (2009). *The probabilistic relevance framework: BM25 and beyond.* Foundations and Trends in Information Retrieval, 3(4), 333–389.

Thomson Reuters. (2023). *CoCounsel: AI legal assistant.* Retrieved March 2026, from https://legal.thomsonreuters.com/en/products/cocounsel

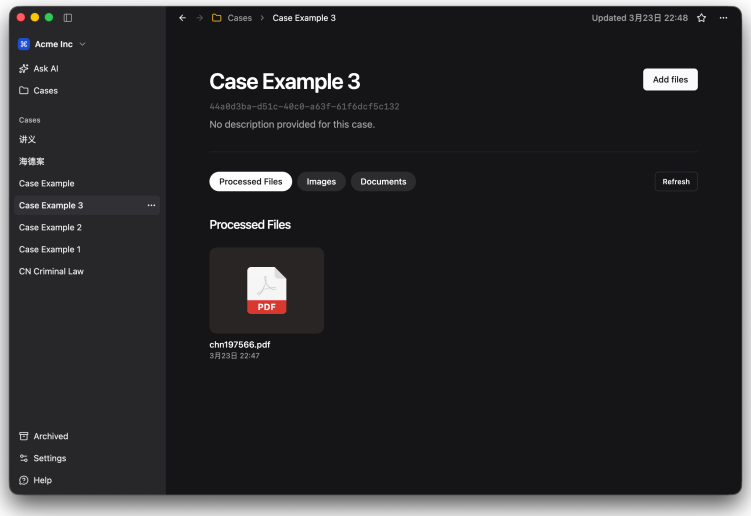
Acknowledgements

The authors would like to thank supervisor Dr. Yue Feng and PhD candidate Ting-Yu Huang for their guidance and valuable insights into RAG implementation throughout this project. We also thank Ms. Shuqi Yang from the *School of Law, Sun Yat-sen University, China* for her insights into legal practice and domain-specific considerations.

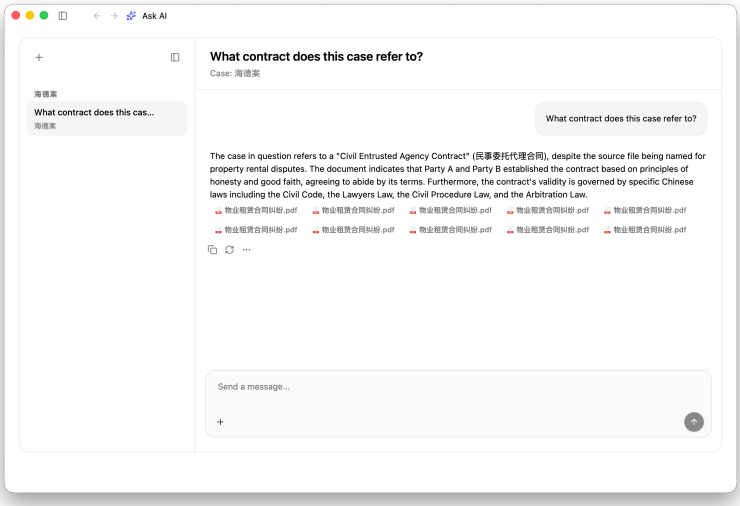
We also acknowledge the open-source community for providing tools and resources that supported the development of this project.

Appendix

Appendix 1. The Demo GUI



Main



Ask AI

Appendix 2. System Prompts

```
system_prompt_keywords_gen = (
  """
  You are an assistant that extracts retrieval keywords from
user queries.
```

```
  Task:
  Extract retrieval terms for BM25.

  Rules:
  1) Return a single field `keywords`.
  2) Include exact names, dates, numbers, citations, section
labels, and identifiers when present.
  3) Use concise retrieval-friendly keywords or short phrases
likely to appear in source documents.
  4) Include formal or domain terms only when clearly supported
by the query.
  5) Do not invent facts or broaden the meaning.
  6) Keep at most {limit} keywords.
  """


SYSTEM_PROMPT = (
      # Role
      "You are a factual legal assistant. "
      # Task
      "Answer only from the provided context. "
      "Do not use outside knowledge. "
      "If the context is insufficient, answer exactly: I don't
have sufficient information to answer this question. "
      "If given a sub-clause, do not assume information from the
main clause that is not explicitly stated in the sub-clause. "
      "Do not invent sources. "
      # Output Format
      "When the context contains enough information, provide a
complete answer rather than an ultra-short label. "
      "Do not mention the context explicitly. "
       "Do not use meta phrases such as 'according to the context'
or 'based on the provided information'. "
      "Write the answer as if directly explaining the situation."
      "If the user asks about a dispute, issue, claim,
allegation, reason, background, or current situation, explain it
in 2-5 sentences using the context. " # Contextual Information
      "Where applicable, include: the nature of the dispute, the
current status, the key claim or issue, and any stated
consequence or requested relief. "
      "Use the same language as the user's question unless the
user explicitly asks for another language. "
      "If the question is in Chinese, answer in Chinese. If the
question is in English, answer in English. " # Language
Specification
      "Prefer a concise explanation over a one-phrase answer."
  )
```

Appendix.3. Pre Process Workflow