Case B – Intelligent Retriever

1. Problem Framing — Blind Spots of Pure Embedding Similarity

Dense vector similarity (FAISS) is powerful but predictable issues appear in long technical/legal documents.

1) Very long documents (300+ pages) but limited top-k retrieval

Blind spot: Retriever keeps returning similar “context-heavy” passages while completely missing rare but important sections.
Example: Many chunks from definitions section retrieved; but the critical “Risk Allocation” clause in the annex is missed.
Reason: Embeddings overweight dense semantic clusters → poor global coverage.

2) Chunk N is retrieved but not N+1 (split clause issue)

Blind spot: When a clause spans multiple chunks, FAISS may return only the first half.
Example: Payment clause 5.1 spans N and N+1. Query retrieves N only → refund conditions in N+1 never seen.
Reason: Similarity is computed per chunk, unaware of adjacency or structural continuity.

3) Query asks for definition of “Term X”, but mentions dominate

Blind spot: System returns multiple “mentions” instead of the one canonical definition.
Example: “Term X” used in examples on pages 10, 50, 200 → these dominate results; actual definition on p.180 is short → lower similarity.
Reason: Dense retrieval prefers longer, richer context passages over short definitions.

2. Signal Design — Making Retrieval Smarter

Instead of relying only on cosine similarity, build a **composite score** mixing multiple signals.

A. Lexical Overlap (BM25 / phrase match)

- **Why:** Exact terms matter for definitions (“Term X”, clause numbers, dates).
- **Metadata needed:** tokenized terms, phrase positions.

B. Structure / Clause Alignment

- **Why:** Clause-specific queries perform poorly without structure info.
- **Metadata:** clause_id, section_id, chunk_index.

C. Provenance Signals

- **Why:** Prevents many hits from the same region; boosts rare sections (annex, schedules).
- **Metadata:** doc_id, page_number, is_annex, rarity_score.

D. Cross-Encoder Reranking Score

- **Why:** Fine-grained semantic relevance; fixes precision issues.
- **Metadata:** reranker_score.

E. Entity / Field Match

- **Why:** If question mentions dates/amounts/parties, prioritize chunks containing those entities.
- **Metadata:** entity lists (NER), numeric fields.

F. OCR Confidence (for scanned PDFs)

- **Why:** Low-quality OCR text should be down-weighted.
- **Metadata:** ocr_confidence.

Combining Signals — Simple Formula

$$
\text{Composite Score} = w_1\text{dense} + w_2\text{lexical} + w_3\text{rerank} + w_4\text{structure} + w_5\text{provenance}
$$

Micro-Example:
Dense=0.55, Lexical=0.70, Rerank=0.92, Structure=1.0 →
Composite score dominated by reranker + structure → correct definition ranked #1.

3. Precision vs Context — Deciding What to Send to the LLM

Different strategies for final chunk selection:

A. Top-k (e.g., k=5)

- **Use when:** Short factual queries (“What is project cost?”).
- **Pros:** Predictable cost.
- **Cons:** Might miss scattered relevant info.

B. Score Threshold (include all > T)

- **Use when:** Recall-critical tasks (legal discovery, auditing).
- **Pros:** High completeness.
- **Cons:** Variable cost.

C. Group Neighbouring / Related Chunks

- **Use when:** Clause spans multiple chunks.

- **Example:** If N selected → also include N±1 within same clause_id.
- **Pros:** Fixes split-clause issue.

D. Summarise or Compress

- **Use when:** Many moderately-relevant chunks found; LLM context is limited.
- **Pros:** Condenses content but keeps meaning.

Decision Rule (simple):

Query Type	Best Strategy
Clause lookup (e.g., 5.2.3)	Neighbouring chunks + top-k
Definition queries	Lexical + structure → small top-k
Open-ended summaries	Threshold + summarise
Legal discovery / compliance	Low threshold (high recall)

4. Evaluation — How to Measure Retrieval Quality (Without Reading Everything)

A. Precision@k

What it measures: how many of top-k chunks are relevant.
If poor:

- *Cause:* Retriever picks context-heavy but irrelevant chunks.
- *Fix:* Increase lexical + reranker weights.

B. Recall@k / Recall@N

What it measures: fraction of all relevant chunks retrieved.
If poor:

- *Cause:* Overemphasis on dense similarity → rare clauses missed.
- *Fix:* Add diversity penalty + section coverage scoring.

C. Hit-Rate / MRR (Mean Reciprocal Rank)

What it measures: how early the first relevant chunk appears.
If poor:

- *Cause:* Definition overshadowed by long mentions.
- *Fix:* Boost definition candidates via structure + lexical signals.

D. Coverage Metric

What it measures: spreads retrieval across document sections.
If poor:

- *Cause:* Redundant, near-duplicate embeddings.
- *Fix:* Use diversity clustering + penalize repeated pages.

E. nDCG@k

What it measures: graded relevance (definition > mention).
If poor:

- *Cause:* Scores not aligned with importance.
- *Fix:* Train a small learned ranker (e.g., LambdaMART).