Grounded Answers from Multi-Passage Regulations: Learning-to-Rank for Regulatory RAG

Tuba Gokhan, Ted Briscoe
Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI)
Correspondence: tuba.gokhan@mbzuai.ac.ae

Abstract

Regulatory compliance questions often require aggregating evidence from multiple, interrelated sections of long, complex documents. To support question-answering (QA) in this setting, we introduce **ObliQA-MP**, a dataset for multi-passage regulatory QA, extending the earlier ObliQA benchmark (Gokhan et al., 2024), and improve evidence quality with an LLM–based validation step that filters out $\sim$20% of passages missed by prior natural language inference (NLI) based filtering. Our benchmarks show a notable performance drop from single- to multi-passage retrieval, underscoring the challenges of semantic overlap and structural complexity in regulatory texts. To address this, we propose a **feature-based learning-to-rank (LTR)** framework that integrates lexical, semantic, and graph-derived information, achieving consistent gains over dense and hybrid baselines. We further add a lightweight score-based filter to trim noisy tails and an obligation-centric prompting technique. On ObliQA-MP, LTR improves retrieval (Recall@10/MAP@10/nDCG@10) over dense, hybrid, and fusion baselines. Our generation approach, based on domain-specific filtering plus prompting, achieves strong scores using the RePAS metric (Gokhan et al., 2025) on ObliQA-MP, producing faithful, citation-grounded answers. Together, **ObliQA-MP** and our validation and RAG systems offer a stronger benchmark and a practical recipe for grounded, citation-controlled QA in regulatory domains.

1 Introduction

Regulatory documents are long, complex, and highly structured, often requiring professionals to synthesize information from multiple sections across documents to answer questions. Unlike typical information retrieval tasks, relevant evidence is frequently distributed across semantically related but non-contiguous passages in multiple documents. This poses challenges for retrieval augmented generative question answering systems: they must identify not only directly answer-bearing text but also supporting context scattered across long complex documents. To support progress in this domain, the Regulatory Information Retrieval and Answer Generation (RIRAG) task (Gokhan et al., 2025) was introduced, along with the ObliQA dataset (Gokhan et al., 2024), which provides synthetic QA pairs grounded in real regulatory obligations. However, the initial formulation of ObliQA primarily focused on single-passage retrieval and relied on limited validation procedures, reducing its applicability to more realistic multi-passage scenarios.

We address these limitations by introducing **ObliQA-MP**, a multi-passage extension of ObliQA requiring evidence from multiple documents, and by refining the validation pipeline. We incorporate a large language model (LLM)–based validation step that significantly improves evidence quality by filtering out approximately 20% of passages incorrectly retained by earlier NLI-based checks. Building on this dataset, we conduct a systematic study of retrieval methods tailored for regulatory texts. Specifically, we explore a feature-based learning-to-rank (LTR) framework that integrates lexical, semantic, and graph-based signals, and evaluate its effectiveness against strong baselines. Our main contributions are:

- **ObliQA-MP**[1]: a multi-passage regulatory QA dataset with LLM-validated question–passage alignment.
- **Feature-based LTR for regulatory retrieval**[2] a framework that combines lexical, semantic, and graph-derived signals, improving over dense and hybrid baselines.

[1] https://github.com/RegNLP/ObliQA-ML/
[2] https://github.com/RegNLP/RegulatoryRAG-ML

Proceedings of the Natural Legal Language Processing Workshop 2025, pages 135–146
November 8, 2025 ©2025 Association for Computational Linguistics

- **Post-retrieval filtering & grounded generation**: a lightweight score-based filter plus an obligation-aware prompting strategy that enforces [P#] citations and provides fallback answers when evidence is insufficient.

2 Related Work

Regulatory and Legal QA Datasets Several datasets support regulatory and legal QA, including EU2UK and UK2EU (Chalkidis et al., 2021b) for legislative alignment, and GDPR-based QA datasets (Abualhaija et al., 2022b,a) for compliance-focused passage retrieval. Broader legal benchmarks like EURLEX (Chalkidis et al., 2021a), LexGLUE (Chalkidis et al., 2022), and ContractNLI (Koreeda and Manning, 2021) focus on classification or entailment tasks. ObliQA (Gokhan et al., 2024) addresses regulatory QA with synthetic question-passage pairs sourced from the financial regulations of a UAE authority. While these resources advance regulatory QA, they primarily focus on single-passage settings and lack fine-grained validation of answer relevance — motivating our work on multi-passage QA with strict evidence alignment in ObliQA-MP.

Retrieval: Sparse, Dense, Fusion, and LTR Classical sparse retrieval (e.g., BM25) remains competitive on legal/regulatory corpora, yet fusion and learning-to-rank (LTR) often yield stronger top-k quality. Reciprocal Rank Fusion (RRF) provides a simple, effective ensemble over heterogeneous runs (Cormack et al., 2009). In-domain dense retrievers (e.g., E5 (Wang et al., 2022)) complement lexical signals, and hybrids (BM25+dense) frequently outperform either alone. Feature-based LTR with LambdaMART (Wu et al., 2010) and LightGBM (Ke et al., 2017) exploits lexical overlap, run scores/ranks, and corpus signals; we adopt a two-stage setup that gathers BM25/dense/RRF candidates and re-ranks with LTR tailored to regulation.

Legal RAG Systems and Benchmarks Legal RAG is emerging, with benchmarks and systems emphasizing evidence-grounded answers (e.g., LegalBench-RAG (Pipitone and Alami, 2024); interpretable statute QA (Louis et al., 2024); case-based and adaptive pipelines (Wiratunga et al., 2024); KG-augmented RAG (Barron et al., 2025)). These works are steps towards practical legal assistants but largely target case law or statutes rather than regulatory obligations. Our focus is complementary: ranking and selecting regulatory passages for grounded answers in a multi-passage multi-document setting.

3 ObliQA-MP: Multi-Passage Dataset for Regulatory QA

We construct **ObliQA-MP**, a multi-passage extension of the ObliQA dataset (Gokhan et al., 2024), starting from the original generation pipeline—which includes structured regulatory documents, topic-based rule clustering, and LLM-based question generation. We merge newly generated multi-passage questions and their associated passages with the original ObliQA dataset, and retain only those questions that are linked to multiple passages, resulting in 13,191 candidate examples.

To ensure the quality and relevance of the supporting evidence, we validate each question-passage pair using the `gpt-4.1-2025-04-14` model. The model receives the following prompt:

```
You are validating if a Passage answers
a Question.
Reply with:
- "Directly Connected": Passage directly
answers.
- "Indirectly Connected": Passage
provides related context.
- "Not Connected": Passage is
irrelevant.
Respond ONLY with JSON: {"Connection":
"...", "ShortReason": "..." }
```

Each passage is labeled according to this schema. Across 13,191 questions and 31,037 passage pairs, 4,212 passages (13.57%) are labeled as *Directly Connected*, 20,474 (65.97%) as *Indirectly Connected*, and 6,351 (20.46%) as *Not Connected*.

Table 1: Distribution of questions in the ObliQA-MP dataset by number of associated passages across train, validation, and test splits.

# Passages	Train	Validation	Test
2 Passages	1,559	322	326
3 Passages	382	93	88
4 Passages	90	16	21
5 Passages	34	9	11
6 Passages	18	6	1
Total	**2,083**	**446**	**447**

To ensure that each retained question is supported by reliable and relevant evidence, we apply two filtering criteria: each question must be associated with at least two passages labeled as either *Directly Connected* or *Indirectly Connected*, and at least one of these passages must be labeled as *Directly Connected*. Applying these criteria removes

3,872 questions that contain fewer than two connected passages and 6,343 questions that lack any *Directly Connected* passage.

The final **ObliQA-MP** dataset consists of 2,976 multi-passage QA pairs. We randomly split this dataset into training, validation, and test sets using a 70/15/15 ratio. Table 1 shows the distribution of questions across splits and the number of associated passages per question. A representative example of a multi-passage QA pair is shown in listing 1.

Listing 1: Example QA pair from ObliQA-MP showing multi-passage evidence across documents.


```
{"QuestionID": "739921c1-385a-4735-a052-
    dee9fba73602",
  "Question": "What are the key
      compliance indicators that a Fund
      Manager should monitor to confirm
      that a Passported Fund is being
      managed and operated within its
      constitutional framework and
      applicable ADGM legislation?",
  "Passages": [
    { "DocumentID": 16,
      "PassageID": "Part 3.6.(2)",
      "Passage": "Each Reporting UAE
          Financial Institution shall
          establish and implement
          appropriate systems and
          internal procedures to enable
          its compliance with the
          Cabinet Resolution and these
          Regulations.",
      "Connection": "Indirectly
          Connected",
      "ShortReason": "The passage
          discusses general
          institutional compliance
          procedures but does not
          directly mention Fund Managers
           or Passported Funds."},
    { "DocumentID": 5,
      "PassageID": "6.1.2",
      "Passage": "The Fund Manager of a
          Passported Fund must:\n(a)
          ensure that the Passported
          Fund is at all times managed
          and operated in compliance
          with its constitution, in
          accordance with applicable
          ADGM legislation, and with
          these Rules; and\n(b) maintain
          , or cause to be maintained, a
           Unitholder register for the
          Passported Fund.",
      "Connection": "Directly Connected
          ",
      "ShortReason": "The passage
          directly outlines the Fund
          Manager's responsibilities
          regarding compliance with the
          constitution and ADGM
          legislation."}
  ]
}
```


The LLM-based validation step introduced in ObliQA-MP is intended to increase the *precision of supervision* by addressing a systematic failure mode of NLI-only validation. In the original ObliQA pipeline, passages were treated as premises and questions as hypotheses under a lightweight NLI model (`nli-deberta-v3-xsmall`) (He et al., 2021). However, NLI-only screening can admit *false positives*: a passage may linguistically entail the question yet fail to provide extractable, citation-backed answer spans (e.g., generic or templated compliance statements)—a problem amplified in multi-passage settings where evidence is distributed. In ObliQA-MP we therefore prompt GPT-4 with citation control to label each candidate passage as *Directly Connected* (contains answer spans), *Indirectly Connected* (supportive context), or *Not Connected*. This finer-grained validation reveals that **20.46%** of the **31,037** evaluated passages are *Not Connected* despite having previously passed NLI filtering. Moreover, only **2,976** out of **13,191** candidate multi-passage questions contain at least one Directly Connected passage and at least one additional supporting passage, underscoring the necessity of stricter validation for dispersed-evidence queries.

As a result, the retained QA pairs are grounded in semantically related passages that also provide *explicit, citation-backed evidence* sufficient to support answer generation; we nonetheless treat the LLM check as a precision-oriented filter rather than expert adjudication and acknowledge that a targeted expert audit remains future work.

4 Feature-Based Learning-to-Rank for Regulatory RAG

Our method builds on the standard RAG pipeline—*retrieve*, *re-rank*, and *generate* (Fig. 1). The key novelty lies in a feature-based learning-to-rank retriever, combined with a lightweight score-based filter to prune noisy candidates before answer generation.

4.1 Retrieval and Ranking Framework

4.1.1 Baseline Retrieval

As baselines, we employ sparse and dense retrievers. For sparse retrieval, we use BM25 (Robertson and Jones, 1976), retrieving the top-k passages ($k \in \{100, 200\}$) per query. While BM25 is robust in capturing keyword overlap, it struggles with

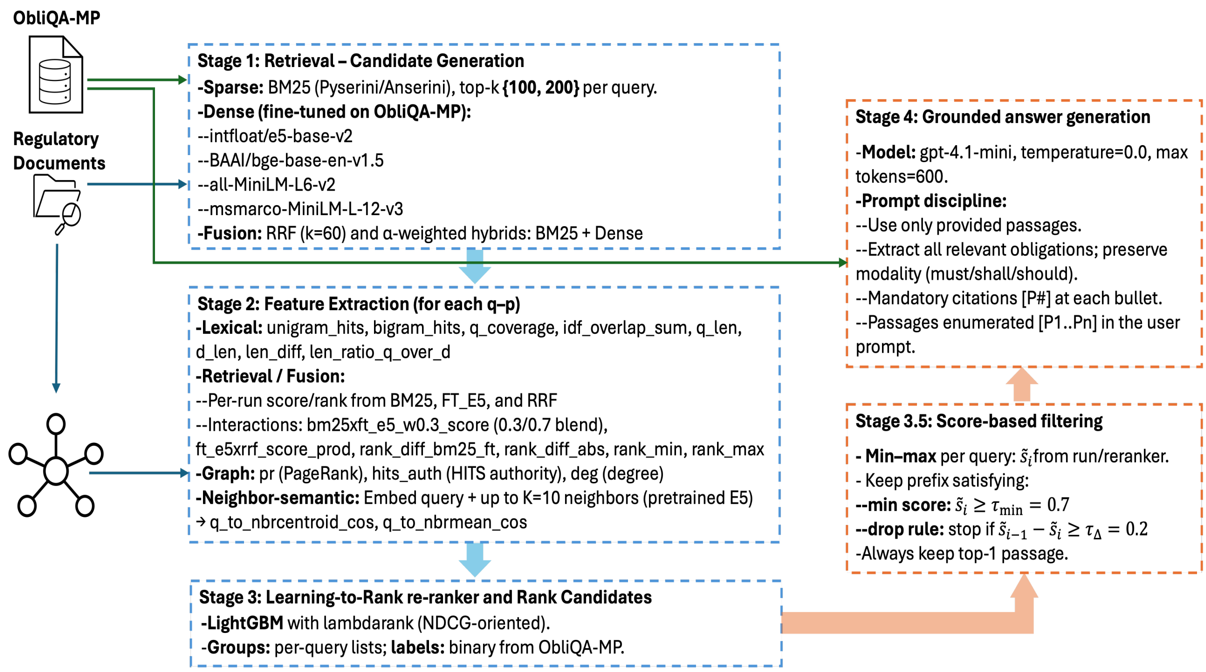

Figure 1: Overview of our retrieval–and–ranking pipeline. Stage 1: candidate retrieval (BM25, fine-tuned dense, and RRF). Stage 2: feature extraction (lexical, retrieval/fusion, graph, and neighbor-semantic features). Stage 3: LTR re-ranking. Stage 3.5: score-based filtering (min–max; thresholds $\tau_{\min}$ and τ_Δ). Stage 4: obligation-centric answer generation with citation control.

paraphrased obligations and multi-passage dependencies. For dense retrieval, we fine-tuned four pretrained encoders on the ObliQA-MP training split: *intfloat/e5-base-v2*, *BAAI/bge-base-en-v1.5*, *all-MiniLM-L6-v2*, and *msmarco-MiniLM-L-12-v3*.

4.1.2 Graph Construction

We represent the corpus as a heterogeneous, directed passage-level graph to encode both the document structure and cross-document references.

Nodes. We create four node types: (1) **Document** nodes for each regulatory document; (2) **Passage** nodes for atomic sections/clauses identified by stable IDs (e.g., "6.1.2"); (3) **NamedEntity (NE)** nodes for globally shared entities extracted from the source JSON (`NamedEntities`); and (4) **DefinedTerm (DT)** nodes for globally shared glossary/definition items (`DefinedTerms`). Each passage node stores the raw text and its document identifier.

Edges. We add typed, directed edges to capture structure and references: (1) **CONTAINS** (*Document* $\rightarrow$ *Passage*) for document membership; (2) **PARENT_OF** (*Passage* $\rightarrow$ *Passage*) for intra-document hierarchy, induced by trimming dot-delimited IDs (e.g., 4.7.14 $\rightarrow$ 4.7); (3) **MENTIONS** (*Passage* $\rightarrow$ *NE*) and **USES_TERM** (*Pas-*

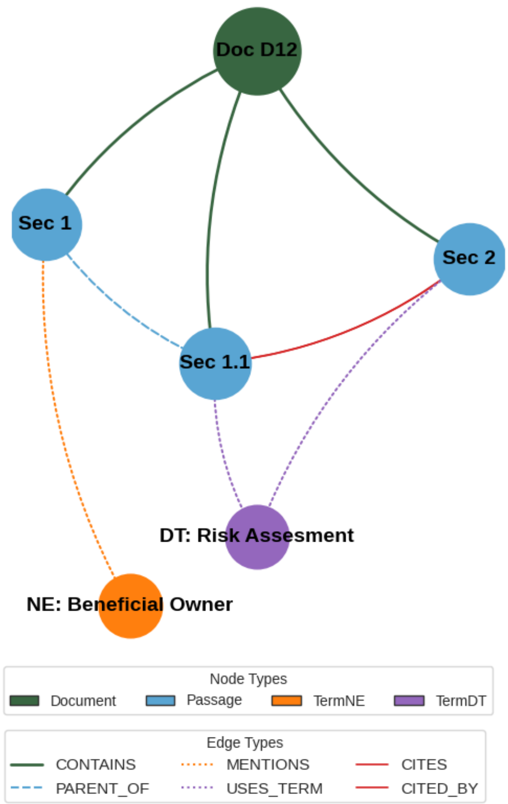

Figure 2: Illustration of the regulatory graph. Documents contain passages; passages cite, mention, and use terms or entities.

Table 2: Feature groups and definitions used in the learning-to-rank model.

Group	Feature	Description
Lexical	**unigram_hits**	Count of overlapping unigrams
	bigram_hits	Count of overlapping bigrams
	q_coverage	Proportion of unique query terms matched in the passage
	idf_overlap_sum	IDF-weighted sum over overlapping terms, $\mathrm{idf}(w) = \log \frac{N+1}{df(w)+1} + 1$
	q_len, **d_len**	Query and passage length (tokens)
	len_diff	Absolute length difference
	len_ratio_q_over_d	Query length divided by passage length
Graph	**pr**	PageRank of the passage node (Brin and Page, 1998)
	hits_auth	HITS authority score (Kleinberg, 1999)
	deg	Degree centrality
Retrieval / Fusion	**bm25_score**, **bm25_rank**	Score and per-query rank from a BM25 run
	ft_e5_score, **ft_e5_rank**	Score and per-query rank from a fine-tuned E5 dense run (base: *intfloat/e5-base-v2*) (Wang et al., 2022)
	rrf_score, **rrf_rank**	Score and per-query rank from precomputed reciprocal rank fusion (Cormack et al., 2009)
	bm25xft_e5_w0.3_score	Linear blend of BM25 and dense scores (0.3 / 0.7)
	ft_e5xrrf_score_prod	Product of dense and RRF scores (if both present)
	rank_diff_bm25_ft	Signed rank difference (BM25 – dense)
	rank_diff_abs	Absolute rank difference
	rank_min, **rank_max**	Min/Max rank across (BM25, dense)
Neighbor-Semantic	**q_to_nbrcentroid_cos**	Cosine between query embedding and centroid of up to K neighbors
	q_to_nbrmean_cos	Mean cosine between query embedding and each neighbor embedding

sage $\rightarrow$ *DT*); and (4) **CITES/CITED_BY** edges between passages based on curated cross-reference tables.

4.1.3 Feature Extraction

For each query–passage pair, we build features that cover lexical overlap, run-based retrieval signals, graph structure, and (optionally) neighbor-aware semantics. Table 2 lists all features.

Lexical and Graph. Lexical overlap features (unigrams, bigrams, IDF-weighted coverage, length statistics) ground the model in surface similarity, while graph centralities (PageRank (Brin and Page, 1998), HITS (Kleinberg, 1999), degree) identify structurally salient passages in the regulatory citation network.

Retrieval / Fusion. We consume scores and ranks from three candidate runs: BM25, a dense retriever fine-tuned from *intfloat/e5-base-v2* on the ObliQA-MP training split, and their Reciprocal Rank Fusion (RRF, $k = 60$) (Cormack et al., 2009). For each run we add {`score`, `rank`} per query. Interaction features include a 0.3/0.7 linear blend (BM25 + dense), a dense$\times$RRF score product, and signed/absolute/min/max rank differences between BM25 and dense.

Neighbor-Semantic. To model local semantic coherence, we embed queries and up to K graph neighbors (K=10) using the *pretrained intfloat/e5-base-v2* encoder (Wang et al., 2022). We then compute cosine similarity to the centroid of neighbor embeddings and the mean over all query–neighbor pairs.

4.1.4 Learning-to-Rank Model

We treat multi-passage selection as a ranking problem and employ a gradient-boosted decision tree framework. We use LightGBM (Ke et al., 2017) with the `lambdarank` objective (Wu et al., 2010), which optimizes Normalized Discounted Cumulative Gain (NDCG) by assigning higher penalties to misordered relevant documents.

Each training instance corresponds to a query–passage pair, represented by the features in Table 2. Candidate sets are formed from the union of BM25, dense, and fused runs, ensuring broad recall. Queries form natural ranking groups, and relevance labels are binary, derived from ObliQA-MP. During training, the model learns feature interactions that discriminate relevant passages from distractors within each query group. At inference time, the trained ranker re-scores the candidate set and outputs a refined ranking. We release all training scripts in our public repository[3].

[3] `https://github.com/RegNLP/RegulatoryRAG-ML`

Table 3: Results of applying the Gokhan et al. (2024) BM25 + GPT baseline from ObliQA to ObliQA-MP.

Dataset	Method	Recall@10	MAP@10	E_s	C_s	OC_s	RePASs
ObliQA	Baseline - BM25(passage)+GPT4	0.761	0.624	0.308	0.123	0.214	0.466
	Baseline - BM25 (fusion)+GPT4	0.764	0.625	0.320	0.131	0.222	0.470
ObliQA-MP	Baseline - BM25(passage)+GPT4	0.561	0.454	0.293	0.129	0.159	0.441
	Baseline - BM25 (fusion)+GPT4	0.561	0.457	0.299	0.145	0.157	0.437

4.2 Post-Retrieval and Answer Generation

Score-Based Filtering. Given a per-query ranked list $(d_1, \ldots, d_K)$ with scores $(s_1, \ldots, s_K)$ from any run or re-ranker, we apply per-query min–max normalization $\tilde{s}_i = \frac{s_i - \min_j s_j}{\max_j s_j - \min_j s_j}$; if $\max_j s_j = \min_j s_j$, we set $\tilde{s}_i = 1$ for all i. We then keep a prefix by two rules applied for $i \geq 2$: (i) **minimum score**: $\tilde{s}_i \geq \tau_{\min}$; (ii) **drop rule**: stop if $\tilde{s}_{i-1} - \tilde{s}_i \geq \tau_\Delta$. We use $\tau_{\min}$=0.7, τ_Δ=0.2, and always keep d_1. This step removes low-utility tail passages while retaining high-confidence evidence for generation (Gokhan et al., 2024).

Answer Generation. We use a deterministic LLM (temperature = 0.0, max output = 600 tokens) with instructions to (i) use only retrieved passages, (ii) extract all obligations relevant to the question, (iii) cite evidence as [P#], and (iv) output *Insufficient evidence in retrieved passages.* if sources are incomplete or contradictory. Concretely, we use `gpt-4.1-mini` with a short, fixed system message and a user prompt that enumerates the filtered passages as [P1..Pn]. See Appendix A for the full prompts and Appendix B for a worked Question/Passages/Answer example.

5 Experiments and Results

5.1 Experimental Setup

We implement all sparse retrieval runs with the `Pyserini` toolkit[4], which provides a standardized interface to Anserini's BM25 implementation. Dense retrievers and cross-encoders are taken from HuggingFace Transformers, and training of the LTR model is carried out with LightGBM's `lambdarank` objective. All feature extraction, training, and evaluation scripts are released in our public repository.[5]

To evaluate retrieval performance, we use the `pytrec_eval` library[6] (Van Gysel and de Rijke, 2018), reporting Recall@10, MAP@10, and nDCG@10. For answer generation, we adopt the **RePASs** metric[7] (Gokhan et al., 2024), a domain-oriented evaluation designed for regulatory QA. Unlike general-purpose metrics such as ROUGE or BLEU, RePASs directly assesses factual grounding and obligation coverage: it combines (i) the average entailment score (E_s), (ii) the average contradiction score (C_s, lower is better), and (iii) the obligation coverage score (OC_s). The final composite captures both semantic faithfulness and domain adequacy.

5.2 Results

Comparative Evaluation with Prior Work To contextualize our results, we reproduce the pipeline of Gokhan et al. (2024), which was originally designed for the ObliQA dataset. Their system combines BM25-based retrieval with GPT-4 for answer generation. For comparability, we re-run their retrieval component on the ObliQA-MP test set and, in the answer generation stage, replace GPT-4 with the more recent `gpt-4.1-2025-04-14`.

As shown in Table 3, BM25 achieves strong results on the original ObliQA dataset (Recall@10 = 0.761) but drops substantially on ObliQA-MP (Recall@10 = 0.561). This degradation illustrates the increased difficulty of multi-passage retrieval and underscores the need for more effective retrieval models in such settings.

Retrieval Results. Table 4 reports retrieval effectiveness across different methods, grouped into baselines, dense retrievers, hybrid combinations, and our proposed LTR models. Among first-stage retrievers, BM25 provides a strong sparse baseline (Recall@10 = 0.549), while dense encoders such as FT-BGE (Recall@10 = 0.573) and FT-E5 (Recall@10 = 0.561) yield modest improvements. Hybrid methods that combine BM25 with dense retrievers consistently outperform individual components, with the best hybrid performance achieved by BM25+FT-BGE at $\alpha = 0.5$ (Recall@10 = 0.617, MAP@10 = 0.480, nDCG@10

[4] https://pypi.org/project/pyserini/
[5] https://github.com/RegNLP/RegulatoryRAG-ML
[6] https://pypi.org/project/pytrec-eval/
[7] https://github.com/RegNLP/RePASs

Table 4: Retrieval performance across different baselines, fusion methods, hybrid approaches, and our Learning-to-Rank (LTR) model on ObliQA-MP test set.

Method	Recall@10	MAP@10	nDCG@10
Baselines			
BM25	0.5493	0.4056	0.5209
Dense (FT_E5)	0.5608	0.3976	0.5096
Dense (FT_BGE)	0.5730	0.4009	0.5143
Dense (FT_MiniLM)	0.5258	0.3698	0.4783
Dense (FT_MSMARCO)	0.5213	0.3582	0.4641
Fusion (RRF)			
RRF(BM25 + FT_E5)	0.6105	0.4714	0.5898
RRF(BM25 + FT_BGE)	0.6173	0.4667	0.5883
RRF(BM25 + FT_MiniLM)	0.5956	0.4474	0.5672
RRF(BM25 + FT_MSMARCO)	0.6066	0.4567	0.5760
Hybrid (BM25 + Dense, α-weighted)			
BM25 + FT_E5 ($\alpha = 0.3$)	0.5977	0.4634	0.5819
BM25 + FT_E5 ($\alpha = 0.5$)	0.6153	0.4888	0.6061
BM25 + FT_E5 ($\alpha = 0.7$)	0.6114	0.4746	0.5910
BM25 + FT_BGE ($\alpha = 0.3$)	0.6000	0.4586	0.5792
BM25 + FT_BGE ($\alpha = 0.5$)	0.6171	0.4795	0.5993
BM25 + FT_BGE ($\alpha = 0.7$)	0.6158	0.4714	0.5894
BM25 + FT_MiniLM ($\alpha = 0.3$)	0.5880	0.4526	0.5719
BM25 + FT_MiniLM ($\alpha = 0.5$)	0.6034	0.4642	0.5841
BM25 + FT_MiniLM ($\alpha = 0.7$)	0.5878	0.4485	0.5655
BM25 + FT_MSMARCO ($\alpha = 0.3$)	0.5917	0.4567	0.5746
BM25 + FT_MSMARCO ($\alpha = 0.5$)	0.6021	0.4701	0.5859
BM25 + FT_MSMARCO ($\alpha = 0.7$)	0.5917	0.4540	0.5671
Hybrid + Secondary Signals (SR)			
BM25 + FT_BGE ($\alpha = 0.5$) + SR(BGE)	0.5174	0.3684	0.4792
BM25 + FT_E5 ($\alpha = 0.5$) + SR(BGE)	0.5245	0.3741	0.4859
BM25 + FT_MiniLM ($\alpha = 0.5$) + SR(BGE)	0.5256	0.3713	0.4838
BM25 + FT_MSMARCO ($\alpha = 0.5$) + SR(BGE)	0.5409	0.3761	0.4920
Learning-to-Rank (Ours)			
LTR	**0.6403**	**0.5116**	**0.6298**

Table 5: Answer generation performance across the best baselines, fusion methods, hybrid approaches, and our Learning-to-Rank (LTR) model on the ObliQA-MP test set.

Method	E_s	C_s	OC_s	RePASs
Gokhan et al. (2024)	0.2990	0.1450	0.1570	0.4370
BM25	0.3916	0.2171	0.1791	0.4512
Dense (FT_BGE)	0.4134	0.1918	**0.2324**	0.4847
RRF(BM25 + FT_BGE)	0.4083	0.2007	0.2164	0.4747
BM25 + FT_E5 ($\alpha = 0.5$)	0.4320	0.1542	0.1960	0.4913
BM25 + FT_MSMARCO ($\alpha = 0.5$) + SR(BGE)	0.4101	0.1785	0.2255	0.4857
LTR	**0.4624**	**0.1340**	0.1984	**0.5090**

= 0.599). RRF also delivers competitive gains, although slightly lower than weighted hybrid fusion.

Our LTR models achieve the strongest performance. In particular, LightGBM trained on the full feature set (`ltr_lgbm_allfeat`) reaches Recall@10 = 0.640, MAP@10 = 0.512, and nDCG@10 = 0.630, outperforming all hybrid and dense-only baselines. This demonstrates the advantage of feature-enriched reranking that integrates lexical, retrieval, graph-based, and neighbor-semantic signals.

Answer Generation Results Our *LTR* is best overall, reaching RePASs 0.50 with the highest E_s (0.46) and lowest C_s (0.13), yielding more faithful and less contradictory answers than all baselines. Relative to the Gokhan et al. (2024) BM25+GPT baseline on ObliQA-MP (RePASs 0.43), this is a

+0.072 absolute gain (+16.5%).

We attribute this lift primarily to **prompt alignment**: the obligation-centric instructions with mandatory [P#] citations (under deterministic decoding) steer the model to extract only supported content, directly improving E_s.

Discussion. Overall, our results demonstrate that LTR models leveraging heterogeneous features provide consistent improvements over both sparse and dense baselines. In particular, the feature-rich LightGBM models outperform fusion strategies such as RRF or linear interpolation, indicating that the model successfully learns how to combine complementary signals.

The comparison with prior work further highlights the limitations of relying solely on BM25 for multi-passage retrieval. While BM25 remains competitive in single-passage ObliQA, its performance degrades sharply in the ObliQA-MP setting. By contrast, our LTR approach maintains strong performance under the more challenging conditions, suggesting better robustness to contextual fragmentation across passages.

These findings confirm that multi-passage regulatory retrieval requires models that integrate structural and semantic cues beyond simple lexical overlap. The integration of graph-derived and neighborhood features is particularly promising, as they allow the retriever to exploit inherent cross-references and local semantic coherence within regulatory corpora. This aligns with our central claim: robust retrieval for complex regulatory tasks demands multi-faceted ranking strategies rather than reliance on single retrieval approaches.

6 Conclusion

We introduced **ObliQA-MP**, a multi-passage regulatory QA dataset, and a stricter LLM-based validation that removes incomplete or off-target evidence more effectively than prior NLI filters. Building on this resource, we presented a *retrieve–re-rank–generate* pipeline that couples a feature-based LTR retriever with lightweight score-based filtering and an obligation-centric prompt with mandatory [P#] citation control.

Our experiments show that multi-passage retrieval is markedly harder than single-passage ObliQA, yet *LTR* consistently outperforms strong sparse, dense, and fusion baselines on retrieval metrics. More importantly, under a common, deterministic generation setup, LTR plus filtering and prompt discipline yields the best answer quality on RePASs—raising entailment and lowering contradictions relative to prior BM25 + GPT baselines on ObliQA-MP.

Future work will focus on exploring hybrid retrievers that combine symbolic and neural representations, developing more robust generation pipelines, and designing evaluation metrics that align more closely with expert-level legal reasoning and compliance practices.

Limitations

ObliQA-MP is a synthetic dataset generated and validated using LLMs, which may introduce linguistic biases and lack the nuance of real-world regulatory queries. While our LLM-based validation improves semantic precision, it cannot fully guarantee legal correctness or reasoning completeness; a targeted expert audit and an error taxonomy remain future work.

The dataset is based solely on regulations from a single regulatory authority, limiting its generalizability to other jurisdictions. Additionally, although each question includes at least one directly connected passage, we do not assess whether all necessary information is present for generating fully comprehensive answers.

Our proposed learning-to-rank framework relies on feature engineering that incorporates lexical, retrieval, graph-based, and neighbor-semantic signals. While this combination provides strong empirical gains, it also increases system complexity and may reduce portability to other domains. The approach requires access to structured cross-references to construct the regulatory graph, which may not be available in less formalized corpora. Moreover, the reliance on pretrained encoders (e.g., *e5-base-v2*) fine-tuned on regulatory QA introduces domain adaptation challenges; performance may vary when applied to corpora with different legal or linguistic characteristics.

Ethics Statement

This work focuses on question answering over publicly available regulatory documents. No personal, sensitive, or proprietary data is used in the construction of the ObliQA-MP dataset. All content is derived from legal and regulatory sources meant for public dissemination. We ensure that all synthetic questions and annotations are automatically generated and validated using LLMs.

References

Sallam Abualhaija, Chetan Arora, and Lionel C. Briand. 2022a. COREQQA: a compliance requirements understanding using question answering tool. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2022, page 1682–1686, New York, NY, USA. Association for Computing Machinery.

Sallam Abualhaija, Chetan Arora, Amin Sleimi, and Lionel C. Briand. 2022b. Automated question answering for improved understanding of compliance requirements: A multi-document study. In *2022 IEEE 30th International Requirements Engineering Conference (RE)*, pages 39–50.

Ryan C Barron, Maksim E Eren, Olga M Serafimova, Cynthia Matuszek, and Boian S Alexandrov. 2025. Bridging legal knowledge and ai: Retrieval-augmented generation with vector stores, knowledge graphs, and hierarchical non-negative matrix factorization. *arXiv preprint arXiv:2502.20364*.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117. Proceedings of the Seventh International World Wide Web Conference.

Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021a. MultiEURLEX - a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6974–6996, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Nikolaos Manginas, Eva Katakalou, and Prodromos Malakasiotis. 2021b. Regulatory compliance through Doc2Doc information retrieval: A case study in EU/UK legislation where text similarity has limitations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3498–3511, Online. Association for Computational Linguistics.

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

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Tuba Gokhan, Kexin Wang, Iryna Gurevych, and Ted Briscoe. 2024. RIRAG: Regulatory information retrieval and answer generation. *Preprint*, arXiv:2409.05677.

Tuba Gokhan, Kexin Wang, Iryna Gurevych, and Ted Briscoe. 2025. Shared task RIRAG-2025: Regulatory information retrieval and answer generation. In *Proceedings of the 1st Regulatory NLP Workshop (RegNLP 2025)*, pages 1–4, Abu Dhabi, UAE. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3149–3157, Red Hook, NY, USA. Curran Associates Inc.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632.

Yuta Koreeda and Christopher Manning. 2021. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Nicholas Pipitone and Ghita Houir Alami. 2024. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain. *arXiv preprint arXiv:2408.10343*.

Stephen E Robertson and K Sparck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146.

Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An extremely fast python interface to trec_eval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 873–876, New York, NY, USA. Association for Computing Machinery.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Nirmalie Wiratunga, Ramitha Abeyratne, Lasal Jayawardena, Kyle Martin, Stewart Massie, Ikechukwu Nkisi-Orji, Ruvan Weerasinghe, Anne Liret, and Bruno Fleisch. 2024. CBR-RAG: Case-based reasoning for retrieval augmented generation in llms for legal question answering. In *Case-Based Reasoning Research and Development*, pages 445–460, Cham. Springer Nature Switzerland.

Qiang Wu, Christopher J. C. Burges, Krysta M. Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Inf. Retr.*, 13(3):254–270.

Appendix

A Answer Generation Prompt

A.1 System Instructions

```
You are a careful compliance QA assistant. Your job
     is to extract and present ALL obligations that
     are directly supported by the retrieved
     passages for the given question.

Rules:
1) Use ONLY the provided passages , no outside
     knowledge or speculation.
2) Include every obligation explicitly supported and
      relevant to the question.
3) Each obligation MUST cite its evidence as [P#] (
     one or more), using only valid passage indices.
4) If the passages are incomplete, ambiguous, or
     contain contradictory obligations relevant to
     the question, reply exactly with: Insufficient
     evidence in retrieved passages.
5) Keep it concise and factual. Use a short bullet
     list; avoid repetition.
6) Preserve modality (must/shall/should); do not
     change meanings.
7) Do not include obligations that are outside the
     scope of the question, even if present in the
     passages.
```

A.2 User Message Template

```
Question:
{<question text>}

Retrieved Passages:
[P1] {<passage 1 text>}
[P2] {<passage 2 text>}
...
[PN] {<passage N text>}

Task: Based ONLY on the passages above, list ALL
     obligations that answer the question. Each
     bullet MUST end with supporting [P#] cites. If
     evidence is insufficient or contradictory,
     reply exactly: 'Insufficient evidence in
     retrieved passages.'
```

B End-to-End Example with Cited Evidence ([P#]): Retrieval → Filtering → Answering

```
Question: What specific actions does the FSRA expect
      an Authorised Person to take to resolve any
     identified breach of Sanctions in a 'timely
     fashion' as mentioned in AML Rule 11.2.1(1)?
Answer:
- An Authorised Person must have powers to resolve
     any breach of sanctions in a timely fashion,
     including taking emergency action itself or
     compelling the delivery and/or storage facility
      to take appropriate action.
- The Authorised Person must have the ability to
     sanction a Member, market participant, or the
     delivery and/or storage facility for acts or
     omissions that compromise compliance with
     applicable sanctions.
- The Authorised Person must notify the Regulator as
      soon as practicable of any breaches or
     potential breaches of applicable sanctions and
     the actions taken to correct them. [P1, P2]

P1: REGULATORY REQUIREMENTS - SPOT COMMODITY
     ACTIVITIES\nSanctions\nPursuant to AML Rule
     11.2.1(1), an Authorised Person must have
     arrangements in place to ensure that only Spot
     Commodities that are not subject to sanctions
     or associated with an entity in the supply
     chain that is itself subject to a sanction, are
      used as part of its Regulated Activities, or
     utilised as part of a delivery and/or storage
     facility operated by itself (or by any third
     parties it uses). In demonstrating compliance
     with the Rule, an Authorised Person must have
     powers to resolve any breach in a timely
     fashion, such as taking emergency action itself
      or by compelling the delivery and/or storage
     facility to take appropriate action. The FSRA
     expects this to include the Authorised Person
     having the ability to sanction a Member, market
      participant or the delivery and/or storage
     facility for acts or omissions that compromise
     compliance with applicable sanctions.
P2: REGULATORY REQUIREMENTS - SPOT COMMODITY
     ACTIVITIES\nSanctions\nPursuant to AML Rule
     11.1.1(2), an Authorised Person must have
     arrangements in place to notify the Regulator,
     as soon as practicable, of any breaches, or
     potential breaches of applicable sanctions, as
     well as the actions it has taken to correct.
     Authorised Persons using Accepted Spot
     Commodities will also need to comply with all
     other applicable FSRA notification requirements
     .


Question: Could you provide further guidance on the
     expectations for an Authorised Person in terms
     of establishing a risk management system that
     addresses the specific risks listed,
     particularly in relation to reinsurance risk?
Answer:
- An Authorised Person must develop, implement, and
     maintain a risk management system to identify,
     assess, monitor, mitigate, and control
     reinsurance risk, which relates to risks from
     the Insurer's use of reinsurance arrangements
     as cedant [P2].
- The risk management system must address all
     material risks, including reinsurance risk
     specifically [P3].
- The system must include a written reinsurance
     management strategy appropriate to the size and
      complexity of the Insurer's operations,
     defining and documenting the Insurer's
     objectives and strategy regarding reinsurance
     arrangements [P4].
- The reinsurance management strategy must include
     systems for selecting reinsurance brokers and
     advisers, selecting and monitoring reinsurance
     programmes, clearly defined managerial
     responsibilities and controls, and clear
     methodologies for determining all aspects of
     the reinsurance programme, including:
    - identification and management of aggregations
         of risk exposure;
    - selection of maximum probable loss factors;
    - selection of realistic disaster scenarios,
         return periods, and geographical
         aggregation areas;
```

```
    - identification and management of vertical and
        horizontal coverage of the reinsurance
        programme [P1].
- The strategy must also include selection of
    participants on reinsurance contracts
    considering diversification and
    creditworthiness, and systems for identifying
    credit exposures to individual reinsurers or
    groups of connected reinsurers on existing
    programmes [P1].

P1: An Insurer's reinsurance management strategy
    should, at a minimum, include the following
    elements:\na.\tsystems for the selection of
    reinsurance brokers and other reinsurance
    advisers;\nb.\tsystems for selecting and
    monitoring reinsurance programmes;\nc.\tclearly
     defined managerial responsibilities and
    controls;\nd.\tclear methodologies for
    determining all aspects of a reinsurance
    programme, including:\ni.\tidentification and
    management of aggregations of risk exposure;\
    nii.\tselection of maximum probable loss
    factors;\niii.\tselection of realistic disaster
     scenarios, return periods and geographical
    aggregation areas; and\niv.\tidentification and
     management of vertical and horizontal coverage
     of the reinsurance programme;\ne.\tselection
    of participants on reinsurance contracts,
    including consideration of diversification and
    credit worthiness; and\nf.\tsystems for
    identifying credit exposures (actual and
    potential) to individual reinsurers or Groups
    of connected reinsurers on programmes that are
    already in place.
P2: An Insurer must develop, implement and maintain
    a risk management system to identify, assess,
    monitor, mitigate and control reinsurance risk.
     Reinsurance risk refers to risks associated
    with the Insurer's use of reinsurance
    arrangements as cedant.
P3: While the risk management systems of an Insurer
    must address all material risks, Rule 2.3 lays
    down specific requirements for an Insurer to
    maintain risk management systems in respect of
    the following areas:\na.\tbalance sheet risk;\
    nb.\tcredit quality risk;\nc.\tnon financial or
     operational risk;\nd.\treinsurance risk; and\
    ne.\tGroup risk.
P4: Without limiting the generality of Rule 2.3.4,
    an Insurer's risk management system in respect
    of its use of reinsurance arrangements must
    include the development, implementation and
    maintenance of a written reinsurance management
     strategy, appropriate to the size and
    complexity of the operations of the Insurer,
    defining and documenting the Insurer's
    objectives and strategy in respect of
    reinsurance arrangements.
```

C Ablation on Learning-to-Rank (LTR)

C.1 Model hyperparameter ablation

We ablate core LTR hyperparameters while holding the feature set in Table 2 and the training protocol fixed. We sweep: (i) candidate-union size $K \in \{100, 200\}$, (ii) LightGBM tree width (`num_leaves` $\in \{63, 127\}$), (iii) learning rate $\in \{0.05, 0.07\}$, and (iv) `min_data_in_leaf` $\in \{50, 100\}$. At test time we re-apply each trained model using the per-model feature order saved at training and the same IDF definition to ensure train/test feature parity. We report nDCG@10, MAP@10, and Recall@10 with `pytrec_eval` on the ObliQA MultiPassage test set.

Results. Table 6 (and the full grid in Table 7) summarizes outcomes. The LTR baseline attains **0.6298** nDCG@10 (MAP@10=**0.5116**, R@10=**0.6403**). Within our grid, the best configuration uses K=200, `num_leaves`= 63, `lr`= 0.05, and `min_leaf`= 50, yielding **0.6121** nDCG@10 (MAP@10=0.4968, R@10=0.6151)—only 0.0177 below the legacy best—indicating the simplified setting remains competitive.

Observations. (1) Larger candidate pools (K=200) consistently help. (2) Slightly smaller trees (`num_leaves`= 63) are marginally preferable to wider ones. (3) A moderate learning rate (0.05) edges out 0.07. (4) Varying `min_data_in_leaf` has a small effect at the top. (5) A lightweight cross-encoder second pass (MiniLM-L-6-v2, w=0.2) did not improve nDCG@10 and is omitted from subsequent results for clarity.

C.2 Feature ablation

We perform a drop-one-group study over the feature groups in Table 2. For each group, we remove its columns from the training/validation feature CSVs, retrain the LTR with the tuned hyperparameters (K=200, `num_leaves`= 63, `lr`= 0.05, `min_leaf`= 50), and evaluate on the same test candidates. Table 8 reports test metrics and absolute deltas w.r.t. the tuned LTR baseline (nDCG@10=0.6121, MAP@10=0.4968, R@10=0.6151).

Results. Dropping *Lexical* features causes the largest degradation (nDCG@10=0.0844, Δ−0.5277; MAP@10=0.0511, Δ−0.4457; R@10=0.1251, Δ−0.4900). Removing *Graph* signals yields nDCG@10=0.5260 (Δ−0.0861), while removing *Neighbor-Semantic* yields nDCG@10=0.5429 (Δ−0.0692). Removing *Retrieval/Fusion* signals results in nDCG@10=0.5717 (Δ−0.0404).[8]

Observations. Lexical matching remains indispensable for this task, with graph-based centrality (PageRank/HITS/degree) and neighbor-semantic cues providing meaningful complementary gains. Fusion signals contribute consistent but smaller improvements than the other groups, suggesting that most of the discriminative power is captured by lexical and graph structure, with fusion providing a pragmatic boost.

[8]In our feature CSVs, *Retrieval/Fusion* corresponds to the precomputed RRF score/rank.

Table 6: LTR baselines on ObliQA MultiPassage (test). We report nDCG@10, MAP@10, and Recall@10 (macro-averaged).

Run (label)	nDCG@10	MAP@10	Recall@10
LTR (all features)	**0.6298**	**0.5116**	**0.6403**
LTR (allfeat, min_leaf=100)	0.6143	0.4959	0.6219
LTR + CE (MiniLM, w=0.2)	0.0504	0.0246	0.0895

Table 7: LTR ablation on ObliQA MultiPassage (test). We vary K, `num_leaves`, learning rate (`lr`), and `min_data_in_leaf`. Neighbor-semantic features are *off* for all rows. Metrics are nDCG@10, MAP@10, Recall@10 (macro-averaged).

K	ns	num_leaves	lr	min_leaf	nDCG@10	MAP@10	Recall@10
200	off	63	0.05	50	**0.6121**	**0.4968**	0.6151
200	off	63	0.07	100	0.6106	0.4955	0.6153
200	off	63	0.05	100	0.6093	0.4935	**0.6164**
200	off	127	0.07	100	0.6075	0.4920	0.6104
100	off	63	0.07	50	0.6075	0.4905	0.6102
200	off	63	0.07	50	0.6069	0.4920	0.6101
100	off	63	0.07	100	0.6069	0.4915	0.6105
100	off	63	0.05	100	0.6066	0.4915	0.6065
100	off	127	0.05	100	0.6063	0.4914	0.6104
100	off	63	0.05	50	0.6063	0.4889	0.6147
200	off	127	0.05	50	0.6060	0.4901	0.6122
100	off	127	0.05	50	0.6053	0.4898	0.6136
200	off	127	0.07	50	0.6040	0.4870	0.6104
100	off	127	0.07	50	0.6037	0.4877	0.6079
100	off	127	0.07	100	0.6028	0.4871	0.6094
200	off	127	0.05	100	0.5981	0.4824	0.6038

Table 8: Feature ablation on the test split. Each row drops one group from the tuned LTR baseline (K=200, ns=off, num_leaves=63, lr=0.05, min_leaf=50). Δ is the absolute difference vs the baseline.

Group	nDCG@10	MAP@10	Recall@10	Δ nDCG@10	Δ MAP@10	Δ Recall@10
Retrieval/Fusion	0.5717	0.4503	0.5900	-0.0404	-0.0465	-0.0251
Lexical	0.0844	0.0511	0.1251	-0.5277	-0.4457	-0.4900
Graph	0.5260	0.4107	0.5257	-0.0861	-0.0861	-0.0894
Neighbor-Semantic	0.5429	0.4250	0.5547	-0.0692	-0.0719	-0.0604