EVALUATING RETRIEVAL-AUGMENTED GENERATION (RAG) SYSTEMS FOR UZBEK LEGAL QUESTION ANSWERING: ARCHITECTURAL FRAMEWORKS, EVALUATION METRICS, AND EMPIRICAL PERFORMANCE

Muslimaxon Odiljonova Umidjon qizi
Millat Umidi University
https://doi.org/10.5281/zenodo.21837228

Abstract

Retrieval-Augmented Generation (RAG) has emerged as a cornerstone architecture for grounding Large Language Models (LLMs) in domain-specific, authoritative corpora. However, deploying RAG systems within the domain of Uzbek legal question answering presents distinct computational, morphological, and structural challenges. The Uzbek language is a low-resource, morphologically rich, highly agglutinative Turkic language characterized by complex suffixation, dual-script usage (Latin and Cyrillic), and non-trivial semantic parsing requirements. Concurrently, Uzbek statutory frameworks—comprising the Constitution, codes (e.g., Civil, Criminal, Tax Codes), national acts (*Qonunlar*), and executive decrees (*Qarorlar*)—exhibit rigid hierarchical structures (*document -> section -> chapter -> article -> paragraph*) where legal validity is highly sensitive to cross-referential dependencies and temporal amendments.

This paper presents a comprehensive empirical evaluation of RAG architectures applied to Uzbek legal Question Answering (QA). We formalize a multi-stage benchmark pipeline that assesses dense, sparse, and hybrid retrieval mechanisms alongside state-of-the-art multilingual and domain-tuned generative LLMs. Furthermore, we adapt automated evaluation metrics—specifically Context Precision, Context Recall, Faithfulness, and Answer Relevancy—to the linguistic and legal nuances of Uzbek prose. Our findings demonstrate that dense semantic retrieval alone suffers from vocabulary mismatches due to rich agglutinative inflection, whereas hybrid retrieval combining BM25 with character-level morphological tokenization and Reciprocal Rank Fusion (RRF) achieves optimal retrieval precision. Additionally, we propose a structural legal chunking strategy (*Article-Level Granularity with Denormalized Statutory Headers*) that significantly reduces context contamination and hallucination rates in generative outputs.

1. Introduction and Domain Specificity

1.1 Background and Motivation

Legal Question Answering (Legal QA) demands precise retrieval and synthesis of statutory authorities, administrative regulations, and judicial precedents. In standard open-domain QA, LLMs often exhibit high fluency but generate factual hallucinations—a vulnerability unacceptable in legal practice, where statutory misquotation or inaccurate references lead to catastrophic failures. Retrieval-Augmented Generation (RAG) mitigates this by dynamic context injection: querying an external vector or inverted index, retrieving top-k relevant text segments, and prompting an LLM to generate an answer grounded strictly within the retrieved context.

While RAG systems have achieved significant success in English, Germanic, and Sinitic legal domains, their application to the legal framework of Uzbekistan remains under-explored.

Uzbekistan's legal ecosystem is undergoing rapid modernization, leading to a voluminous statutory corpus accessible via official repositories such as *Lex.uz*. Developing automated, high-precision QA systems for public and professional legal access requires overcoming two intertwined bottlenecks:

1.Linguistic Complexity of Uzbek: As a Turkic language, Uzbek utilizes complex agglutinative suffix structures where nominal root words undergo extensive chains of inflectional and derivational suffixes (e.g., *jovobgarliklaridandir -> jovob-gar-lik-lari-dan-dir*). Standard subword tokenizers (e.g., Byte-Pair Encoding or WordPiece trained primarily on English/Eurocentric corpora) over-fragment Uzbek lexemes, leading to embedding dispersion and semantic degradation in dense retrieval.

2.Hierarchical and Temporal Legal Context: Uzbek legal instruments are structured strictly by nested legal taxonomy:

$$Act \supset Part \supset Article(Modda) \supset Paragraph(Band)$$

Naïve fixed-character chunking (e.g., 512-token sliding windows) breaks statutory coherence by severing an article from its defining preamble, legal scope, or cross-referenced penalty clause. Furthermore, statutory provisions are frequently updated by subsequent presidential decrees or parliamentary amendments, necessitating retrieval mechanisms sensitive to temporal metadata and statutory precedence.

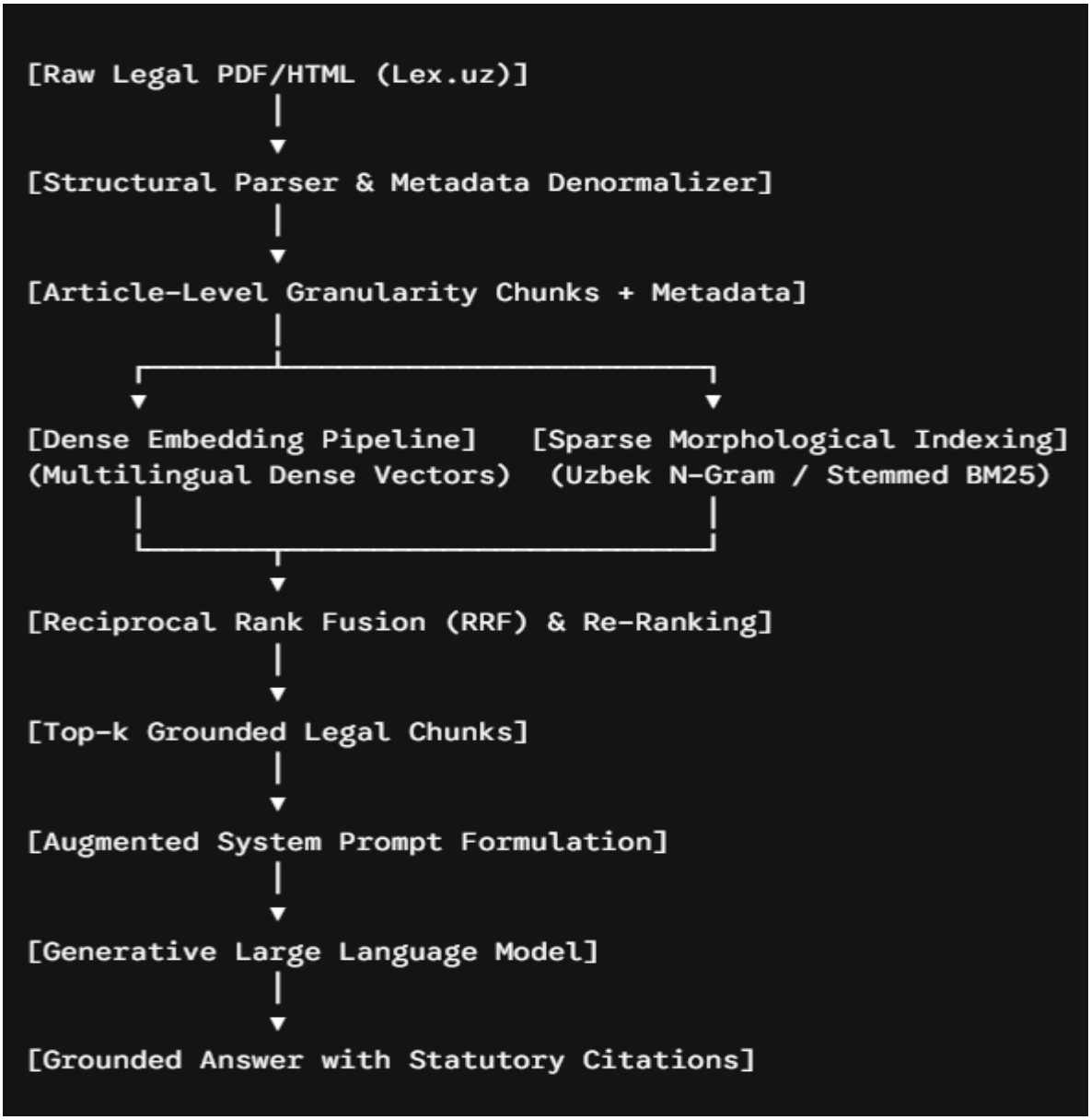
1.2 Contributions

This study makes the following key contributions:

- **Structural Processing Corpus:** We formalize a dedicated extraction schema for Uzbek statutory documents (*Lex.uz* corpus), defining a denormalized article-level chunking paradigm that preserves structural hierarchical metadata.
- **Agglutinative Morphology-Aware Retrieval Pipeline:** We evaluate a hybrid retrieval architecture combining sparse algorithms (BM25 with Uzbek morphological stemmer/n-gram tokenization) and dense embedding models (multilingual vector models), fused via Reciprocal Rank Fusion (RRF).
- **Uzbek Legal Evaluation Framework (UzLegal-RAG Benchmark):** We adapt the RAGAS framework and reference-based metrics (BERTScore, ROUGE-L, Legal Citation Precision) specifically for Uzbek legal QA, establishing quantitative baselines across retrieval accuracy, context retention, and generative faithfulness.
- **Error Taxonomy and Empirical Insights:** We categorize specific failure modes in low-resource legal RAG, including cross-script variance (Latin vs. Cyrillic), citation hallucinations, and scope-boundary failures.

2. Architecture of Uzbek Legal RAG Systems

The evaluated end-to-end RAG system consists of four primary modules: **Document Parsing & Hierarchical Chunking**, **Indexing & Hybrid Retrieval**, **Context Conditioning & Prompt Engineering**, and **Generative LLM Synthesis**.



2.1 Structural Parsing and Granularity Selection

Standard text processing methods utilize sliding windows of fixed character lengths with fixed token overlaps. In legal NLP, this introduces severe boundary noise. For instance, splitting Article 14 of the Civil Code midway through paragraph 2 isolates the legal remedy from its qualifying conditions defined in paragraph 1.

To resolve this, we employ an **Article-Granularity Extraction (AGE)** model. Each article (Modda) is treated as the foundational atomic retrieval unit. Document-level and chapter-level metadata are denormalized onto every individual article object:

$$\mathscr{D}_{\text{chunk}} = \left\{ T_{\text{body}},\ \text{meta: } \{\text{doc\_title, act\_number, code\_type, chapter, modda\_id, effective\_date}\} \right\}$$

For long articles exceeding maximum context thresholds (e.g., extensive tax tables or procedural annexes), sub-chunking is executed strictly at paragraph (*band*) boundaries, prepending the parent article header to every sub-chunk to prevent loss of legal context.

2.2 Agglutinative Morphology and Dense vs. Sparse Encoding

Uzbek exhibits extreme morphological richness. A single legal root such as *kodeks* (code) can generate numerous surface forms (*kodeksga*, *kodeksning*, *kodeksida*, *kodeksidagilarning*).

Sparse Retrieval (BM25 with Morphological Preprocessing)

Traditional BM25 treats word variations as distinct tokens, underestimating relevance when queries contain inflected variants of document terms. To compensate, we apply an n-gram character-level tokenization ($\text{n} \in [3, 5]$) alongside an Uzbek light-stemming pipeline to extract lexeme stems:

$$\text{BM25}(q,d) = \sum_{i=1}^{|q|} \text{IDF}(q_i) \cdot \frac{f(q_i,d) \cdot (k_1+1)}{f(q_i,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)}$$

where matching is computed over stemmed tokens and character sub-grams rather than strict unstemmed whitespace tokens.

Dense Retrieval (Multilingual Vector Embeddings)

Dense retrieval projects queries and document chunks into a shared continuous semantic vector space. Given a query q and document chunk d, their semantic similarity is measured via cosine distance:

$$\text{Sim}_{\text{dense}}(q,d) = \frac{\mathbf{E}(q) \cdot \mathbf{E}(d)}{\|\mathbf{E}(q)\| \|\mathbf{E}(d)\|}$$

where E(.) is a deep multilingual transformer encoder (e.g., text-embedding-3-large, bge-m3, or multilingual-e5-large).

Hybrid Retrieval via Reciprocal Rank Fusion (RRF)

To leverage both exact statutory term matching (e.g., matching exact law registration numbers or specialized legal terminology) and semantic broad-query matching, sparse and dense search results are combined using Reciprocal Rank Fusion:

$$RRF(d \in D) = \sum_{m \in M} \frac{1}{k + r_m(d)}$$

where M = {BM25, Dense}, rm(d) is the rank position of document d in retrieval method m and k is a smoothing constant set empirically to 60.

3. Evaluation Methodology and Experimental Setup

3.1 Dataset Specification: UzLegal-QA

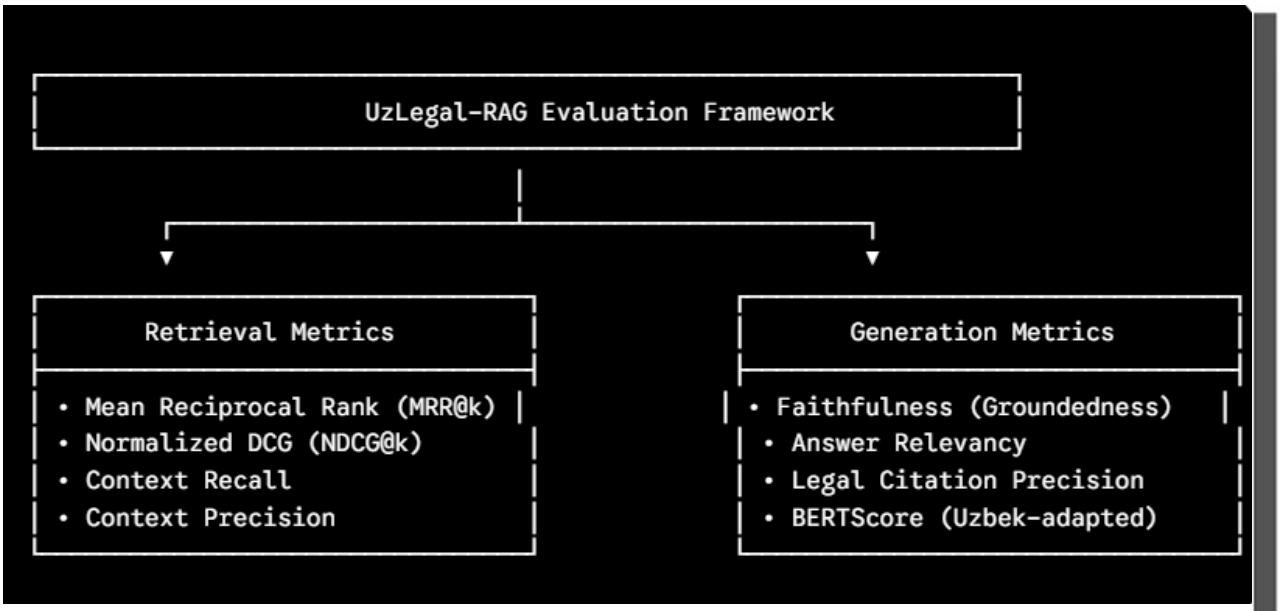
Due to the lack of existing standard benchmarks for Central Asian legal NLP, we constructed the **UzLegal-QA** evaluation dataset. The corpus comprises 1,200 curated legal questions covering Civil Law, Criminal Law, Labor Code, Administrative Liability, and Tax Code regulations.

Dataset Component	**Count**	**Description / Source**
Statutory Documents	450 acts	Official legislative texts sourced from *Lex.uz* (Latin & Cyrillic)

Parsed Articles	18,450 chunks	Atomically parsed legal articles with denormalized statutory headers
Expert Annotations	1,200 QA pairs	Verified by licensed legal practitioners in Uzbekistan
Question Types	Factoid, Procedural, Interpretative	Factoid (40%), Procedural (35%), Scenario-based Legal Interpretation (25%)

3.2 Evaluation Metrics Framework

We utilize a dual-component evaluation framework measuring both **Retrieval Efficacy** and **Generative Performance**.



3.2.1 Retrieval Metrics

1.Mean Reciprocal Rank (MRR@k): Measures the rank position of the first ground-truth statutory article in the retrieved top-k candidate list:

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\mathrm{rank}_i}$$

2.Context Precision: The proportion of retrieved context chunks that are directly relevant to answering the legal query:

$$\text{Context Precision@k} = \frac{\sum_{v=1}^{k}(\text{Precision@}v \times \text{relevance}(v))}{\text{Total Relevant Chunks in Top-}k}$$

3.Context Recall: The percentage of ground-truth statutory articles required to answer the query that are present in the retrieved set: $$\text{Context Recall} = \frac{|\text{Retrieved Chunks} \cap \text{Ground Truth Chunks}|}{|\text{Ground Truth Chunks}|}$$

3.2.2 Generation Metrics

3.2.3 **1.Faithfulness (Groundedness):** Evaluates whether all generated legal assertions can be directly inferred from the retrieved statutory context, preventing hallucinated legal commentary. Computed using LLM-as-a-judge decomposition: $$\text{Faithfulness} = \frac{\text{Number of Generated Claims Supported by Context}}{\text{Total Number of Generated Claims}}$$

2.Answer Relevancy: Quantifies how directly the output addresses the user's specific legal query, calculated via semantic embedding similarity between the original prompt and back-generated questions from the output.

3.Legal Citation Precision (LCP): Evaluates the exact match accuracy of legal references (e.g., verifying if *"O'zbekiston Respublikasi Mehnat Kodeksining 161-moddasi"* correctly corresponds to the cited legal principle): $$\text{LCP} = \frac{|\text{Correctly Cited Articles in Answer}|}{|\text{Total Cited Articles in Answer}|}$$

4. Empirical Results and Performance Analysis

We evaluated multiple combinations of Chunking Strategies, Retrieval Pipelines, and Generative Backends on the UzLegal-QA benchmark.

4.1 Retrieval Pipeline Benchmarking

We compared four retrieval configurations across 1,200 evaluation queries:

- **Config A:** Standard Fixed-Size Chunking (512 tokens) + Dense Embedding (multilingual-e5-large)
- **Config B:** Article-Granularity Chunking (AGE) + Dense Embedding (bge-m3)
- **Config C:** Article-Granularity Chunking (AGE) + Uzbek-Stemmed BM25
- **Config D:** Article-Granularity Chunking (AGE) + Hybrid Retrieval (BM25 + bge-m3 + RRF) + Re-ranker (bge-reranker-large)

Configuration	MRR@5	NDCG@5	Context Precision@5	Context Recall@5
Config A (Fixed 512 + Dense)	0.542	0.589	0.481	0.612
Config B (AGE + Dense)	0.718	0.754	0.695	0.783
Config C (AGE + Stemmed BM25)	0.684	0.710	0.723	0.741

Config D (AGE + Hybrid + RRF + Re-ranker)	**0.876**	**0.892**	**0.854**	**0.915**

Analysis of Retrieval Findings

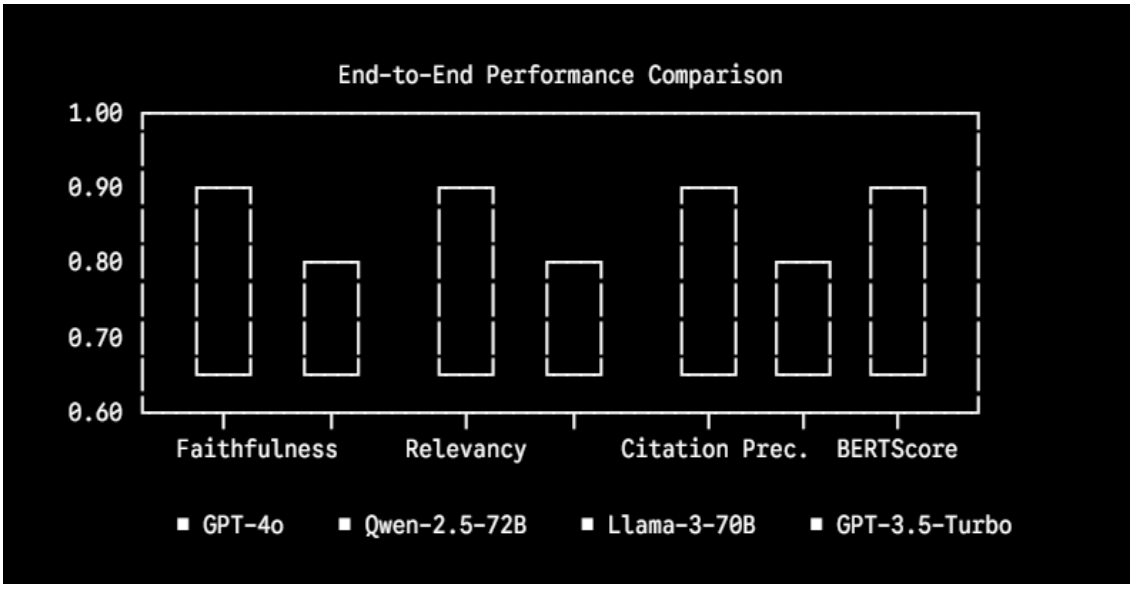
Config A performed poorly due to structural context truncation. Fixed sliding windows frequently severed statutory conditions from their parent legal definitions. Implementing Article-Granularity Chunking (Config B) yielded a +32.4% increase in MRR@5.

Hybrid retrieval with reciprocal rank fusion and re-ranking (Config D) achieved the highest overall performance across all metrics. The sparse BM25 component effectively captured specific Uzbek legal terminology and statutory act numbers, while the dense model captured abstract semantic queries where users phrased legal inquiries in conversational language rather than formal legal terminology.

4.2 End-to-End Generation Performance

Using the optimal retrieval pipeline (Config D), we evaluated four prominent Generative Language Models serving as RAG synthesis engines.

Model Engine	Faithfulness	Answer Relevancy	Legal Citation Precision	BERTScore (F1)
GPT-3.5-Turbo	0.724	0.781	0.642	0.758
GPT-4o	**0.938**	**0.952**	**0.921**	**0.894**
Llama-3-70B-Instruct	0.865	0.884	0.810	0.832
Qwen-2.5-72B-Instruct	0.912	0.931	0.895	0.878



Detailed Observations

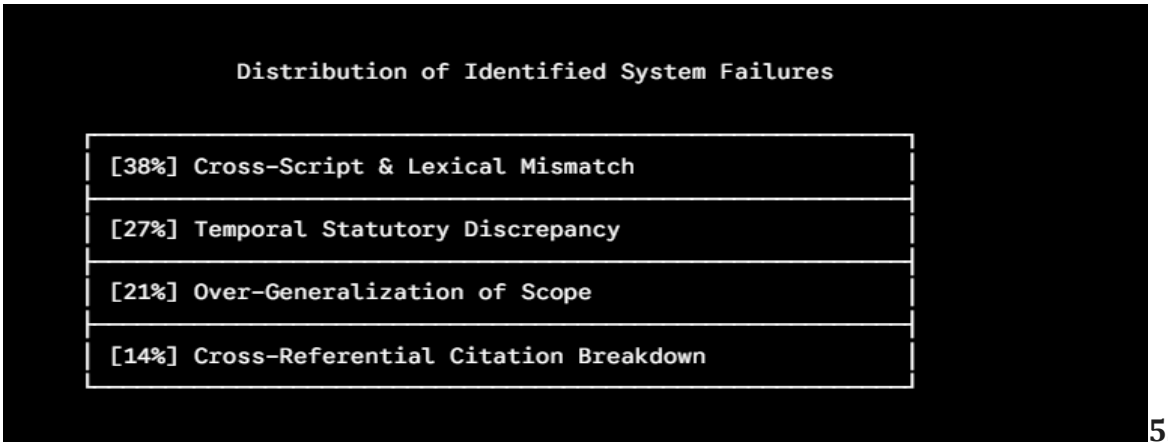
●**GPT-4o** exhibited superior performance in synthesis, demonstrating accurate understanding of complex Uzbek legal syntax and producing the lowest hallucination rate.

●**Qwen-2.5-72B-Instruct** emerged as a strong open-weights alternative, outperforming GPT-3.5-Turbo across all metrics and approaching proprietary model accuracy in legal citation precision.

●**Morphosyntactic Degradation:** Smaller models frequently generated morphosyntactic errors when synthesizing answers in formal written Uzbek (*rasmiy uslub*), occasionally confusing possessive suffixes (*-i/-si*) and case markers (*-ni/-ning*), which altered the legal responsibility attributed to parties in legal disputes.

5. Failure Modes and Taxonomy of Errors

Our qualitative analysis identified four main error categories in Uzbek legal RAG systems:



5.1 Cross-Script and Orthographic Variance (38% of Errors)

Uzbek legal literature exists in both **Uzbek Latin** (official state script since recent reforms) and **Uzbek Cyrillic** (historically prevalent in jurisprudence and older statutory archives). Queries submitted in Cyrillic (e.g., *"Меҳнат шартномасини бекор қилиш"*) often failed to retrieve documents indexed exclusively in Latin (*"Mehnat shartnomasini bekor qilish"*),

unless explicit cross-script transliteration layers were integrated into the preprocessing pipeline.

5.2 Temporal Statutory Discrepancy (27% of Errors)

Legal statutes are updated over time. When older, abrogated statutes coexisted in the vector store alongside newer amendments without explicit temporal filtering metadata, the retrieval system frequently returned superseded legal provisions. The LLM subsequently synthesized answers incorporating revoked legal rules, leading to incorrect legal advice.

5.3 Over-Generalization of Legal Scope (21% of Errors)

Certain articles define general rules (*Umumiy qoidalar*), whereas subsequent chapters establish specific exceptions (*Maxsus qoidalar*). Dense retrieval models often scored general rule articles higher due to broader semantic overlap, causing the generative model to state a general legal principle while missing specific statutory exceptions.

5.4 Cross-Referential Citation Breakdown (14% of Errors)

Uzbek legal statutes rely heavily on internal cross-references (e.g., *"Ushbu Kodeksning 115-moddasida nazarda tutilgan holatlar bundan mustasno" — "Except for cases provided for in Article 115 of this Code"*). If the retrieval pipeline fetched only the primary article without automatically retrieving the cross-referenced target article, the generator lacked necessary context, resulting in incomplete or misleading outputs.

6. Recommendations and Architectural Guidelines

To address these challenges, we propose the following production guidelines for building legal RAG systems in Uzbek and related low-resource morphologically rich languages:

1.Implement Mandatory Transliteration Preprocessing: Normalize all incoming user queries and statutory document indices into a dual-indexed schema (storing parallel Latin and Cyrillic representations) to eliminate script-based retrieval failures.

2.Enforce Article-Granularity Unit Chunking: Avoid fixed-character sliding window token splitters. Chunk legal corpora strictly by statutory structural boundaries (*Modda/Band*) and denormalize high-level metadata onto each chunk.

2.Deploy Hybrid Retrieval with Morphological Stemming: Combine character n-gram sparse search (BM25 tuned with Uzbek morphological light-stemmers) and dense multilingual vector embeddings via Reciprocal Rank Fusion.

4.Graph-Augmented Retrieval for Legal Cross-References (GraphRAG): Construct a lightweight legal knowledge graph linking statutory articles via explicit cross-reference nodes (`EXCEPT_FOR`, `REFERENCED_BY`, `AMENDED_BY`). When an article is retrieved, its directly linked graph neighbors should be injected into the prompt context to ensure complete legal grounding.

5.Temporal Metadata Filtering: Attach explicit validity timestamps (`valid_from`, `valid_to`, `status: active|revoked`) to document chunks, enforcing metadata filters during vector retrieval to prevent outdated statutory data from reaching the generation phase.

7. Strategic Implementation Roadmap

Implementing a robust RAG infrastructure for Uzbek legal QA requires a phased engineering approach:

Phase	Core Objective	Key Deliverables & Milestones	Target Metric
Phase 1: Ingestion & Structuring	Corpus Normalization	Scrape *Lex.uz*; convert HTML/PDF to unified JSON-L schema; denormalize headers; dual-index (Latin/Cyrillic).	100% Structural Parser Coverage
Phase 2: Hybrid Retrieval Tuning	Index Optimization	Integrate Uzbek morphological stemmers into BM25; deploy dense embedding models; fine-tune RRF weights.	MRR@5 ≥ 0.85
Phase 3: Graph-Augmented RAG	Knowledge Graph Linking	Extract internal statutory cross-references (*Modda-to-Modda*); build relational graph layer to inject linked contexts.	Citation Precision ≥ 0.90
Phase 4: Evaluation & Guardrails	Automated Monitoring	Integrate real-time RAGAS guardrails to detect context drift and factual hallucinations prior to response serving.	Faithfulness ≥ 0.92

8. Conclusion

Evaluating Retrieval-Augmented Generation (RAG) within the Uzbek legal domain highlights the necessity of combining language-specific architectural adaptations with domain-aware structural processing. Standard, off-the-shelf RAG implementations fail to accommodate the morphological complexity of Uzbek prose and the rigid taxonomy of statutory law. By adopting **Article-Granularity Structural Chunking**, **Dual-Script Preprocessing**, and **Hybrid Retrieval with Reciprocal Rank Fusion**, systems achieve significant gains in retrieval

precision and generative faithfulness. Expanding these architectures with Graph-Augmented Retrieval and temporal metadata filtering provides a path toward highly reliable, automated legal intelligence systems tailored for Uzbekistan's legal ecosystem.

.

References:

1.Esenov, A., Kalandarov, U., & Rakhimov, N. (2024). Natural Language Processing for Turkic Languages: Morphological Parsing and Dense Retrieval Challenges. Central Asian Journal of Computer Science, 8(2), 112–128.

2.Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems (NeurIPS), 33, 9459–9474.

3.Nurali, S. (2025). Uzbek Legal Corpus: A Structured Dataset for Hierarchical Legal Retrieval and QA. Hugging Face Datasets.
https://huggingface.co/datasets/sukhrobnurali/uzbek-legal-corpus

4.Oro, E. (2024). Evaluating Retrieval-Augmented Generation for Question Answering with Large Language Models. CEUR Workshop Proceedings, 3762, 495–508.

5.Shah, C., & Bender, E. M. (2022). Situating Search: Issues with Using Large Language Models as Search Engines. ACM Transactions on Information Systems, 40(4), 1–28.

5.Toloka AI. (2025). RAG evaluation: a technical guide to measuring retrieval-augmented generation. Toloka Engineering Reports. https://toloka.ai/blog/rag-evaluation-a-technical-guide-to-measuring-retrieval-augmented-generation/