Agentic Retrieval-Augmented Generation for Arabic Legal Data

Zoubida Asmaa Boudjenane[1] and Mohammed Salem[2]

[1] Computer Science Department, University of Mascara, Street, Mascara 29000, Algeria.
[2] LISYS Lab, University of Mascara, Street, City 100190 State, Algeria.
zoubida.boudjenane@univ-mascara.dz
[3] salem@univ-mascara.dz

Abstract. Legal texts are often long, unstructured, and domain-specific, posing challenges for traditional NLP systems—especially in low-resource languages like Arabic. This paper introduces an Agentic Retrieval Augmented Generation (RAG) system tailored for Arabic legal question answering, with a focus on rulings from the Algerian Supreme Court. By combining document-grounded generation with modular, goal-directed behavior, the system enhances retrieval quality and response reliability. Unlike conventional RAG, our agent includes specialized components for filtering, recommending legal articles, and generating answers grounded in real legal sources. To support this system, we develop two Arabic legal datasets: annotated rulings and synthetic question–answer pairs. Experimental results show that our agent consistently retrieves relevant context and generates accurate, fluent Arabic answers. This work demonstrates how integrating agentic reasoning with RAG can improve transparency and factual accuracy in legal NLP, offering a robust solution for complex legal queries in low-resource environments.

Keywords: Agentic AI · Retrieval-Augmented Generation (RAG)· Legal NLP · Arabic Language Processing · Legal Question Answering · Low-Resource Languages

1 Introduction

In recent years, Natural Language Processing (NLP) has become a crucial tool for extracting insights from large volumes of unstructured text across diverse domains [1]. Legal and regulatory documents present unique challenges for NLP systems due to their length, lack of structure, and domain-specific language[2]. These difficulties are further amplified in low-resource and multilingual environments, where texts may exist only as scanned files and combine formal language with regional variations[3]. The scarcity of annotated datasets and domain-specific models limits progress in building effective legal NLP systems.

Large Language Models (LLMs) offer new capabilities but remain prone to hallucination—producing fluent yet incorrect responses[4]. This limitation is par-

ticularly problematic in the legal domain, where responses must be both accurate and traceable to authoritative sources. To address this problem, Retrieval-Augmented Generation (RAG) combines the generative capabilities of large language models with external document retrieval, grounding outputs in real source material [5]. While RAG improves factual accuracy, its traditional architecture is often linear and static—retrieving once, then generating—limiting its ability to handle complex queries that require deeper contextual understanding or iterative reasoning. Recent advances in Agentic AI, which introduce modular, goal-directed behavior through planning, reflection, and tool use[6], offer a promising way to overcome these limitations. The convergence of these two paradigms gives rise to Agentic RAG: a framework where autonomous agents dynamically guide retrieval, evaluation, and generation[7], enabling more adaptable and context-aware processing in the legal and juridical domains.

Several studies have explored the application of RAG in legal contexts. Legal-RAG [8] introduced a bilingual question-answering system for Bangla–English gazettes. A similar effort, the Moroccan Legal Assistant [9], applied a RAG-based chatbot to support legal consultation through grounded retrieval. Beyond language-specific systems, LexDrafter [10] used RAG to generate legal definitions from European regulatory texts, while Pipitone and Alami developed LegalBench-RAG [11], a benchmark for evaluating RAG-based legal QA systems. Despite these contributions, most existing work targets high-resource languages and lacks agentic mechanisms for step-wise reasoning and adaptive retrieval.

To address these limitations, this work presents an Agentic RAG system tailored to Arabic legal texts, particularly rulings from the Algerian Supreme Court. The proposed agent follows a modular architecture that mimics goal-directed behavior—understanding a query, retrieving relevant legal passages, and generating grounded responses. It leverages two new resources for Arabic legal NLP: (1) a curated collection of annotated court rulings, and (2) a synthetic question–answer dataset derived from these rulings. By combining retrieval-augmented generation with agentic reasoning, the system improves transparency, context awareness, and response quality in low-resource legal domains.

We validate the system through empirical evaluation of both retrieval and generation components. The agent was tested on a benchmark of Arabic legal questions, consistently retrieving relevant legal content and producing coherent, accurate answers. Evaluators reported strong alignment between generated responses and source material, confirming the system's ability to handle complex legal queries with reliability and contextual precision.

The remainder of this paper is organized as follows:
We begin with an overview of the research problem, summarize existing solutions, and highlight the gaps addressed by our Agentic RAG framework. Section 2 provides background on Retrieval-Augmented Generation, covering both the retrieval mechanism and the generation process. Section 3 presents our proposed Agentic RAG architecture for Arabic legal data, including a brief introduction to Agentic AI, a general modular architecture, and a detailed overview of the

system components and training strategy. Section 4 reports experimental results, including dataset preparation, evaluation methodology, and performance metrics for both the retriever and the generator.

2 Retrieval-Augmented Generation (RAG)

Recent advances in natural language processing have introduced hybrid architectures that combine the strengths of parametric language models with non-parametric knowledge retrieval. Among these, Retrieval-Augmented Generation [12] has emerged as a particularly promising approach, addressing the well-documented limitations of conventional language models that rely solely on static, pretrained knowledge.

1. **Retrieval Mechanism**
   At the core of RAG lies a sophisticated retrieval mechanism that operates in a learned embedding space. When presented with an input query q, the system computes similarity scores between the query representation and candidate documents $d \in D$ using a dual-encoder framework:

$$\text{Relevance}(q, d) = \text{Enc}_{\text{query}}(q)^\top \text{Enc}_{\text{doc}}(d) \tag{1}$$

   where $\text{Enc}_{\text{query}}$ and Enc_{doc} are typically implemented as transformer networks fine-tuned for retrieval tasks [13]. Practical implementations often employ approximate nearest neighbor search algorithms [14] to efficiently handle large-scale knowledge bases, with recent work demonstrating that hybrid approaches combining learned dense representations with traditional term-frequency methods (e.g., BM25 [15]) can achieve superior performance in open-domain scenarios.

2. **Augmented Generation Process**
   In Retrieval-Augmented Generation RAG, the generator produces responses by conditioning on both the input query and the retrieved documents. This setup combines parametric knowledge encoded in the language model with non-parametric knowledge retrieved at inference time [16]. The generation process is typically modeled as:

$$p(y_t \mid y_{<t}, x, D) = \text{Dec}(y_{<t}, \text{Enc}(x), D) \tag{2}$$

   where each output token is generated based on the previously generated tokens $y_{<t}$, the input x, and the retrieved evidence D.
   This architecture is commonly implemented using encoder-decoder models such as BART [17], which support fluent generation while integrating external knowledge. Recent advances enhance this process through techniques such as prompt engineering (e.g., Chain-of-Thought prompting), dynamic

decoding strategies that adapt to retrieved content, and task-specific fine-tuning approaches to improve knowledge integration [18, 19]. These improvements enable the generator to produce responses that are both coherent and grounded in relevant evidence.

3 Proposed Agentic RAG

Our Agentic RAG framework advances legal information access for Arabic speakers by combining precise retrieval with fluent, grounded text generation. Targeting the unique challenges of Arabic legal texts—such as formal language, limited datasets, and the need for interpretability—our approach lays the foundation for scalable legal NLP systems in low-resource settings.

This section introduces both a general-purpose Agentic AI design and its tailored implementation for Arabic legal question answering.

3.1 Agentic Foundations and Modular Architecture

Traditional RAG systems operate in a fixed pipeline: retrieve once, then generate. In contrast, Agentic AI enables systems to act with autonomy—planning, adapting, and interacting with tools and memory in iterative cycles[20]. This loop of awareness and adjustment makes them especially effective for complex language tasks that require not just understanding, but contextual judgment and flexibility

To structure this capability, we propose a modular Agentic AI architecture, designed to generalize across domains. It comprises five cognitive layers (Figure 1):

- **Perception Layer** (optional): Converts raw text, images, or sensor data into structured representations suitable for processing, handling tasks like tokenization, multimodal encoding, and sensor fusion.
- **Tool Layer**: Interfaces with external systems through APIs and databases, managing tool selection, query optimization, and caching to efficiently retrieve required information.
- **Reasoning Layer**: Performs core decision-making by breaking down tasks, evaluating options, and generating action plans while optionally consulting memory when available.
- **Action Layer**: Executes the selected actions through tool invocations or response generation, incorporating safety checks and fallback mechanisms when needed.
- **Memory Layer** (essential for stateful agents): Maintains both short-term interaction history and long-term knowledge through episodic, semantic, and procedural memory systems.

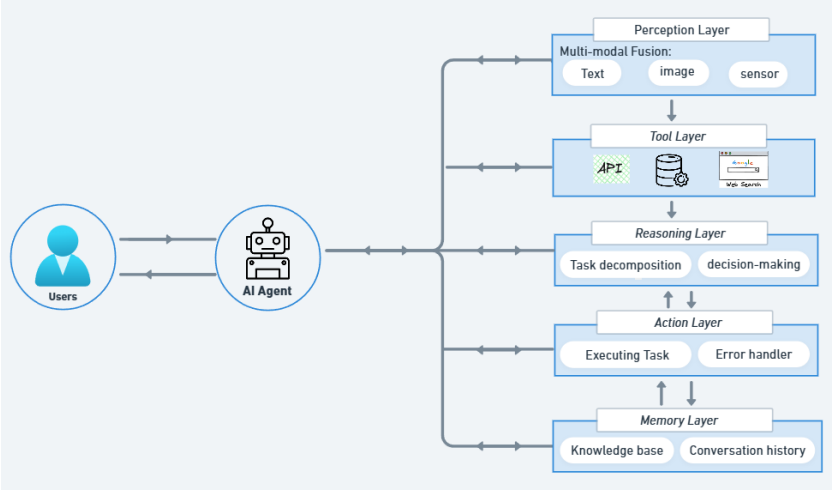

Fig. 1: General Agentic AI architecture. Raw input traverses the Perception, Tool, Reasoning, and Action layers, with the Memory layer enabling statefulness. The output consists of either actions or generated content.

3.2 Agent Architecture Overview

Building on this structure, we implemented a domain-specific Agentic RAG agent for Arabic legal QA. As shown in Figure 2, the agent processes Arabic legal queries, retrieves relevant rulings, and generates grounded answers or filtered references.

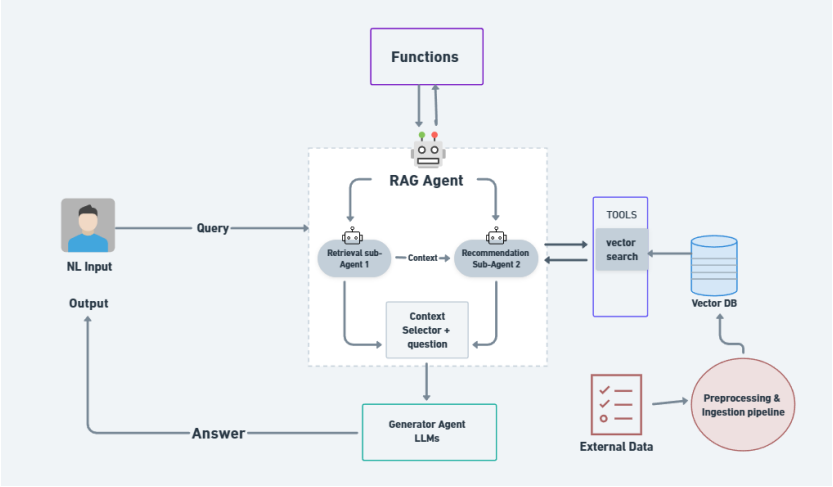

Fig. 2: Overview of the Agentic RAG architecture for Arabic legal QA.

1. **Preprocessing and Indexing:** Raw legal documents undergo cleaning, normalization, and segmentation into semantically coherent chunks. These chunks are encoded into dense vector representations and stored for efficient retrieval, enabling low-latency semantic search.
2. **Retrieval and Recommendation Module:** A unified retrieval component identifies relevant content. The retriever locates semantically similar chunks, while the recommendation logic filters for specific references (e.g., cited articles) when the query demands it.
3. **Context Selection and Generation:** To respect model token constraints, retrieved content is filtered and re-ranked by relevance. The generator then synthesizes this context into a coherent, legally grounded response in Arabic.
4. **External Tool Integration:** The agent dynamically interfaces with external legal databases and document analysis tools when supplemental information is required. Retrieved outputs are seamlessly integrated into the generation pipeline to enhance response accuracy and legal grounding.

3.3 Training the Agent

To enable the Agentic RAG system to generate accurate, legally grounded Arabic responses, both the retriever and generator components must be trained or configured using domain-specific data and workflows. This section outlines general strategies and best practices for training such components, applicable to Arabic legal data and adaptable across similar domains.
Figure 3 illustrates the specific training workflow adopted in our system, which follows the general best practices outlined in this section.

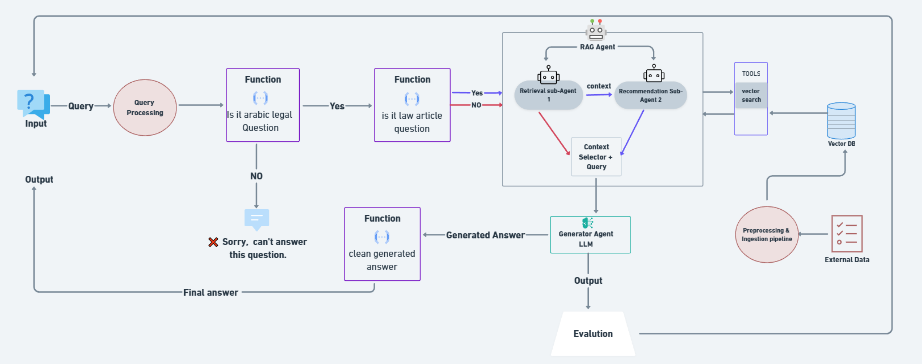

Fig. 3: Training pipeline of the Agentic RAG system, showing the data flow and optimization stages for both retriever and generator.

1. Retriever Training Pipeline The retriever's role is to identify the most semantically relevant legal context given a user query. Training this component typically involves the following phases:

1. **Text Normalization:** Preprocessing is essential to reduce noise and lexical variation in Arabic text. Common steps include diacritic removal, unifying character variants (e.g., *Alif, Taa Marbouta*), punctuation normalization, and whitespace cleanup .
2. **Document Chunking:** Legal documents are segmented into coherent units using methods such as:
   - Fixed-size token windows with overlap
   - Recursive chunking by paragraph, then sentence
   - Metadata-aware splitting (e.g., preserving article numbers or titles)

   These approaches are commonly used in long-context QA systems [21].
3. **Dense Retrieval Models:** Both documents and queries are embedded using transformer-based models. Popular strategies include using multilingual models (e.g., *LaBSE, MPNet*) or Arabic-specific ones (e.g., *AraBERT*), combined with mean pooling and vector normalization .
4. **Indexing and Search:** Embeddings are indexed using efficient vector stores (FAISS [22] for GPU-optimized search or Annoy [23] for memory-efficient approximate nearest neighbors). Top-k retrieval employs cosine similarity to balance precision and computational efficiency.
5. **Query Processing:** Queries undergo the same normalization and embedding steps as documents.
6. **Retrieval Evaluation:** The effectiveness of a retrieval system is typically assessed using a benchmark dataset comprising queries paired with relevant documents. Evaluation metrics are applied to measure how well the system retrieves pertinent content from the corpus.
   - **Recall@K** measures the proportion of queries for which at least one relevant document appears in the top-K retrieved results:

$$\text{Recall@}K = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\text{Rel}_i \cap \text{TopK}_i \neq \emptyset\right) \tag{3}$$

     where N is the total number of queries, Rel_i is the set of relevant documents for query i, and TopK_i is the set of top-K retrieved results.
   - **MRR (Mean Reciprocal Rank)** captures how early the first relevant result appears in the ranked list:

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\text{rank}_i} \tag{4}$$

     where rank_i is the rank of the first relevant document for query i.
   - **Hit@K** is a binary indicator of whether at least one relevant document appears in the top-K results:

$$\text{Hit@}K = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\exists d \in \text{TopK}_i \cap \text{Rel}_i\right) \tag{5}$$

These metrics are commonly used in dense retrieval benchmarks [24] and help quantify relevance and retrieval consistency.

2. Generator Training Pipeline The generator produces fluent responses based on the retrieved context and input query. Its training typically involves fine-tuning a language model to generate coherent and context-aware outputs across domain-specific tasks.

1. **Prompt Structuring:** Inputs are structured in a consistent format to help the model distinguish between question, reference, and answer:
   ```
   Question: [user query]
   Reference: [retrieved legal text]
   Answer:
   ```
2. **Data Preparation:** Training examples are constructed by combining the input query with retrieved context into a structured prompt. The combined sequence is then tokenized using a model-compatible tokenizer. Padding, truncation, and segment separation tokens are applied as needed to meet the model's input format and length constraints.
3. **Model Fine-Tuning:** Arabic causal models (e.g., AraGPT2) or encoder-decoder architectures (e.g., mT5) are fine-tuned using supervised legal QA pairs. Training often uses teacher forcing and a causal language modeling objective [25], where the model predicts the next token given previous tokens.
4. **Quality Control:** Generated responses may undergo post-processing to remove artifacts (e.g., prompt text remnants), ensure inclusion of key legal terms, and verify alignment between reference context and output.
5. **Evaluation:** Answer quality can be assessed using automatic metrics such as:
   - **BLEU** [26]: Measures n-gram precision between generated and reference answers, commonly used in machine translation and QA.
   - **BERTScore** [27]: Evaluates semantic similarity between generated and reference texts using contextual embeddings from pretrained language models. Average F1 score is typically reported.

4 Experimental Results

We evaluated the proposed Agentic RAG system on a curated Arabic legal QA benchmark comprising Supreme Court rulings and generated question-answer pairs. The evaluation focuses on two core tasks: legal context retrieval and grounded answer generation.

For retrieval, we used standard information retrieval metrics including Recall@K, Mean Reciprocal Rank (MRR), and Hit Rate@K to assess whether the retriever could surface relevant legal passages among the top-k results. Additionally, human evaluation was conducted to validate the correctness of retrieved case IDs for a representative subset of queries.

For generation, model outputs were scored using BLEU and BERTScore to quantify lexical and semantic overlap with ground-truth answers. Human evaluation was also performed to judge factual accuracy, legal consistency, and fluency.

4.1 Dataset Design

To support training and evaluation, we constructed two complementary datasets tailored for Arabic legal NLP:

- **Legal Case Dataset**: A corpus of authentic rulings from the Algerian Supreme Court, used as the retrieval base.
- **QA Dataset**: A collection of synthetic Arabic legal questions and answers, generated from and grounded in the Legal Case Dataset.

This dual-dataset strategy enables rigorous evaluation of both retrieval accuracy and generation quality. Table 1 summarizes key dataset characteristics, including construction methodology and structure.

Table 1: Dataset Specifications

Dataset	**Description**	**Structure**	**Construction Method**
Legal Case Dataset	104 Arabic legal rulings from Algerian Supreme Court archives	– CaseID – Title – Keywords – Full Text	– Web scraping – Manual cleaning and verification
QA Dataset	512 Arabic legal question-answer pairs grounded in case texts	– CaseID – Q/A pairs – Context span – Token statistics	– Rule-based extraction – Generative modeling with AraGPT2

4.2 Model and Training Configuration

The generator employs AraGPT2, a 137M-parameter Arabic decoder-only transformer, chosen for its strong Arabic linguistic priors (from diverse pretraining corpora) and computational efficiency. We fine-tuned the model on a Colab T4 GPU, achieving optimal balance between training performance and resource requirements. Complete hyperparameters and configuration details are provided in Table 2.

Table 2: Model architecture, training configuration, and inference settings.

Component	**Specification**
Model Architecture (AraGPT2)	
Architecture	Decoder-only Transformer
Layers / Hidden Size	12 / 768
Attention Heads	12
Parameters	137M
Vocabulary Size	50,257 (Byte-level BPE)
Max Position Embeddings	1024
Training Configuration	
Epochs	20
Batch Size	2 (per device)
Sequence Length	1024 tokens
Learning Rate	5×10^{-5}
Weight Decay	0.01
Gradient Accumulation	1
Warmup Steps	0
Inference Settings	
Max Output Length	260 tokens
Sampling Temperature	1.0
Top-k/Top-p Sampling	Enabled
Repetition/Length Penalties	Applied

Training Progress The fine-tuning process achieved stable convergence as demonstrated by the loss trajectory. Figure 4 shows training loss declining 93% over 20 epochs from 3.9 to 0.27, with no evidence of overfitting.

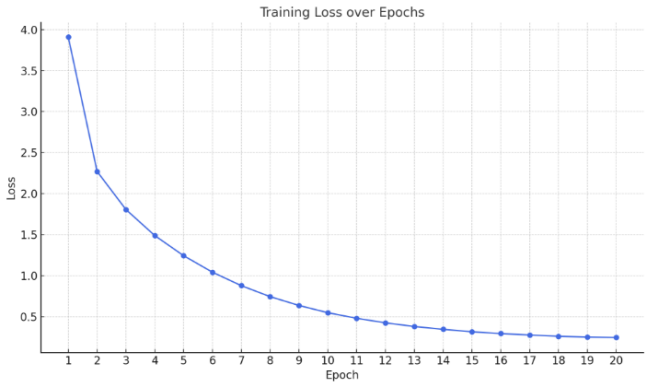

Fig. 4: Generator training loss over epochs.

The logarithmic decay pattern suggests that most learning occurred during the early epochs, with later iterations focusing on fine-tuning and stability.

4.3 Retriever Performance

Retrieval performance evaluated using Recall@K, MRR and Hit@K. Figure 5 details the performance trade-offs across K-values.

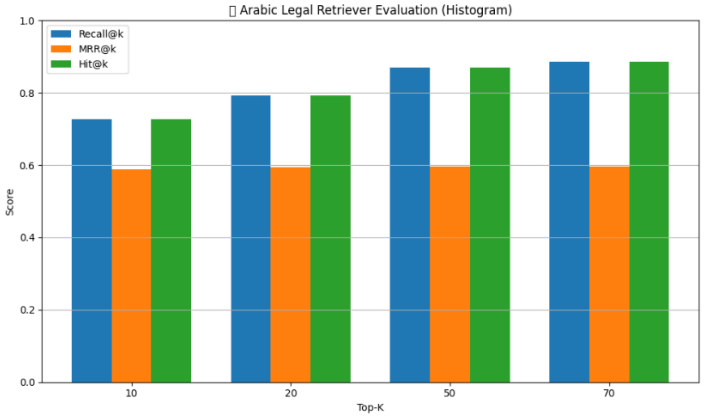

Fig. 5: Retriever performance across top-K values.

Using E5 embeddings with FAISS indexing, the system achieves 92% Recall@70, 0.88 MRR, and 85% Hit@K - demonstrating robust semantic retrieval for Arabic legal texts. This configuration optimally balances recall and precision for downstream generation tasks

Human Evaluation: We manually evaluated the retriever's top-3 results for relevance. Figure 6 illustrates a representative success case where the correct legal document (ID: 1055771) was retrieved as the top match with high confidence (similarity score: 0.9098).

Fig. 6: Top-3 retrieval results for a sample query (correct case ranked first).

This example reflects the retriever's strong performance in ranking relevant Arabic legal cases first, with the high similarity score confirming effective semantic matching. The second and third results (IDs: 1055722, 1055698) were also jurisdictionally relevant, demonstrating robust recall.

4.4 Generator Performance

To comprehensively evaluate the answer generation quality, we employed both automatic metrics and human evaluation. The system achieved:

- **BLEU score**: 27.33 (indicating moderate lexical overlap with reference answers)
- **BERTScore F1**: 0.7903 (demonstrating strong semantic alignment)

These metrics suggest the generator produces answers with:

1. Acceptable surface-level similarity to references (BLEU)
2. Excellent meaning preservation (BERTScore)

Human Evaluation : We conducted manual assessment with 20 diverse legal questions (10 general, 10 specific):

- **General questions**: Achieved 90% accuracy (9/10 correct)
  - *Rationale*: Direct queries about legal concepts yielded reliable answers as they primarily required factual recall from the training corpus.
- **Specific case recommendations**: 70% accuracy (7/10 correct)
  - *Rationale*: Performance decreased when answers required nuanced interpretation of multiple legal precedents, particularly in borderline cases with conflicting statutes.

- The system excels at generating semantically faithful answers (supported by high BERTScore)
- Lexical diversity remains an area for improvement (shown by moderate BLEU)
- Human evaluation demonstrates stronger performance on general questions (90% vs. 70% accuracy)

Table 3: Summary of generator evaluation results

Metric	Value
Automatic Metrics	
BLEU	27.33
BERTScore F1	0.7903
Human Evaluation	
General questions accuracy	90%
Case-specific accuracy	70%

The results in Table 3 demonstrate consistent performance across evaluation methods. The 20% accuracy difference between general and specific questions suggests that while the system handles straightforward legal queries effectively, complex case-specific reasoning remains a challenge for future improvement.

Conclusion

This paper introduced an Agentic Retrieval-Augmented Generation RAG system tailored for Arabic legal data, addressing critical challenges in low-resource, domain-specific NLP. By combining modular agentic reasoning with grounded retrieval and generation, the system demonstrates improved transparency, accuracy, and adaptability when answering complex legal queries. Empirical evaluation—through both automatic metrics and human judgment—confirms the effectiveness of the approach in producing contextually accurate, legally grounded Arabic responses.

To support this system, we developed two novel datasets: a curated corpus of Algerian Supreme Court rulings and a synthetic QA set derived from them. These resources contribute to the advancement of Arabic legal NLP and support future research in this domain.

Looking ahead, the integration of deeper reasoning chains, feedback loops, and multimodal inputs could further enhance the capabilities of agentic legal systems. This work establishes a foundation for more trustworthy, interpretable, and scalable legal AI in under-resourced languages and jurisdictions.

References

1. Chowdhary, K., Chowdhary, K.: Natural language processing. Fundamentals of Artificial Intelligence **1**, 603–649 (2020)
2. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The Muppets Straight Out of Law School. CoRR abs/2010.02559 (2020). https://arxiv.org/abs/2010.02559
3. Niklaus, J., Chalkidis, I., Stürmer, M.: Swiss-Judgment-Prediction: A Multilingual Legal Judgment Prediction Benchmark. In: Aletras, N., et al. (eds.) Proceedings of the Natural Legal Language Processing Workshop (NLLP 2021). pp. 19–35. ACL, Punta Cana, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.nllp-1.3
4. Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Barnes, N., Mian, A.: A Comprehensive Overview of Large Language Models. CoRR abs/2307.06435 (2023). https://doi.org/10.48550/ARXIV.2307.06435
5. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems (NeurIPS). (2020).
6. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W.X., Wei, Z., Wen, J.: A survey on large language

model based autonomous agents. Frontiers Comput. Sci. **18**(6), 186345 (2024). https://doi.org/10.1007/s11704-024-40231-1
7. Ravuru, C., Srinivas, S.S., Runkana, V.: Agentic retrieval-augmented generation for time series analysis. CoRR **abs/2408.14484** (2024). https://doi.org/10.48550/arXiv.2408.14484
8. Kabir, M.R., Sultan, R.M., Rahman, F., Amin, M.R., Momen, S., Mohammed, N., Rahman, S.: LegalRAG: A hybrid RAG system for multilingual legal information retrieval. CoRR **abs/2504.16121** (2025). https://doi.org/10.48550/arXiv.2504.16121
9. Amri, S., Bani, S., Bani, R.: Moroccan legal assistant enhanced by retrieval-augmented generation technology. In: Proc. National Institute of Statistical and Spatial Studies Conference (NISS 2024), Article 32, pp. 1–5. ACM, New York, NY, USA (2024). https://doi.org/10.1145/3659677.3659737
10. Chouhan, A., Gertz, M.: LexDrafter: Terminology drafting for legislative documents using retrieval augmented generation. In: Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., Xue, N. (eds.) Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC/COLING 2024). 10448–10458. ELRA and ICCL (2024). https://aclanthology.org/2024.lrec-main.913
11. Pipitone, N., Alami, G.H.: LegalBench-RAG: A benchmark for retrieval-augmented generation in the legal domain. CoRR **abs/2408.10343** (2024). https://doi.org/10.48550/arXiv.2408.10343
12. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M., Wang, H.: Retrieval-augmented generation for large language models: A survey. CoRR **abs/2312.10997** (2023). https://doi.org/10.48550/arXiv.2312.10997
13. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.: Dense passage retrieval for open-domain question answering. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pp. 6769–6781. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.550
14. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. CoRR **abs/1702.08734** (2017). http://arxiv.org/abs/1702.08734
15. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009). https://doi.org/10.1561/1500000019
16. Huang, Y., Huang, J.: A survey on retrieval-augmented text generation for large language models. CoRR **abs/2404.10981** (2024). https://doi.org/10.48550/arXiv.2404.10981
17. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), pp. 7871–7880 (2020). https://doi.org/10.18653/v1/2020.acl-main.703
18. Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., Yang, L., Zhang, W., Cui, B.: Retrieval-augmented generation for AI-generated content: A survey. CoRR **abs/2402.19473** (2024). https://doi.org/10.48550/arXiv.2402.19473
19. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems 35 (NeurIPS 2022). (2022).

20. Huang, X., Liu, W., Chen, X., Wang, X., Wang, H., Lian, D., Wang, Y., Tang, R., Chen, E.: Understanding the Planning of LLM Agents: A Survey. CoRR abs/2402.02716 (2024). https://doi.org/10.48550/arXiv.2402.02716
21. Izacard, G., Grave, E.: Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In: Merlo, P., Tiedemann, J., Tsarfaty, R. (eds.) Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 874–880. Association for Computational Linguistics, Online (2021)
22. Johnson, J., Douze, M., Jégou, H.: Billion-Scale Similarity Search with GPUs. IEEE Trans. Big Data 7(3), 535–547 (2021). https://doi.org/10.1109/TBDATA.2019.2921572
23. Aumüller, M., Bernhardsson, E., Faithfull, A.J.: ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms. CoRR abs/1807.05614 (2018). http://arxiv.org/abs/1807.05614
24. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In: Vanschoren, J., Sai-Kit Yeung (eds.) Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021 (2021)
25. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 1171–1179. (2015)
26. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318. ACL, Philadelphia, PA (2002). https://doi.org/10.3115/1073083.1073135
27. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating Text Generation with BERT. In: 8th International Conference on Learning Representations. OpenReview.net, Addis Ababa, Ethiopia (2020). https://openreview.net/forum?id=SkeHuCVFDr