

JUDICIAIRE: A LEGAL AI CHATBOT FOR MOROCCAN LAW USING DEEP LEARNING MODELS

El Kahlaoui Youssef ¹, Essafi Anass ², Gorry Ayoub ³

Cherradi Mohammed ⁴

Digital Transformation and Artificial Intelligence

National School of Applied Sciences of Al Hoceima

Thursday 22, June 2025

Abstract- JudiciAire is a specialized legal chatbot focused on Moroccan law, developed using large language models (LLMs) and parameter-efficient fine-tuning (PEFT) techniques to deliver accurate and accessible legal assistance. Trained on a curated dataset of digitized Moroccan legal codes sourced from official government platforms, the system addresses the challenge of limited computational resources by fine-tuning the Mistral-7B-Instruct v0.3 model through two approaches: a general multi-domain model with limited batches and a more focused model trained specifically on the Moroccan Family Code (Moudawana) with extended epochs. The latter proved more effective in citation accuracy and numerical precision. The chatbot is deployed via a secure, user-friendly web application built with a React frontend and Flask backend, using Clerk for authentication and MongoDB for data storage. JudiciAire demonstrates the potential of deep learning in legal tech and provides a replicable blueprint for developing domain-specific AI tools, offering insights applicable to other legal systems and specialized knowledge domains.

Index Terms- Legal Chatbot, Moroccan Law, Deep Learning, Large Language Models (LLMs), LLM Fine-Tuning, Mistral, Code de la Famille, Natural Language Processing (NLP), Retrieval Augmented Generation (RAG), Unsloth, PDF Data Extraction, OCR, Generative AI, Gemini, Qdrant, Resource-Constrained AI Development, Bilingual Data Processing.

Table of Contents:

1. Introduction:	3
2. Related Work:	3
3. Methodology:	4
3.1 Data Collection and Processing:	4
3.2 Model Selection and Fine-tuning:	5
3.3 Training Process and Resource Constraints:	6
3.4 Q&A Pair Generation:	8
4. Web Application Development:	9
4.1 Architecture:	9
4.2 Frontend Implementation	9
4.3 Backend Infrastructure	10
4.4 User Authentication and Data Persistence	10
5. Experimental Results	11
6. Discussion	13
7. Conclusion	13
8. Acknowledgements	14
9. References	14
10. Appendices:	14
References	15
Authors	16

1. INTRODUCTION:

Access to legal information and expertise remains a significant challenge across many jurisdictions, creating barriers for individuals seeking to understand their rights and obligations. In Morocco, as in many countries, legal codes and regulations are often complex, difficult to navigate, and not easily accessible to the general public. The JudiciAIre project aims to bridge this gap by creating an AI-powered legal assistant specifically trained on Moroccan legal texts.

The advent of large language models (LLMs) presents an opportunity to democratize access to legal information through conversational AI interfaces. However, generic LLMs often lack the specialized knowledge and context required for legal advice, particularly within specific jurisdictions. This research addresses this limitation through domain-specific fine-tuning on a corpus of Moroccan legal texts.

The objectives of this project are to:

1. Create a specialized dataset of Moroccan legal codes and jurisprudence
2. Fine-tune a large language model on legal domain knowledge
3. Develop a user-friendly web interface for legal inquiries
4. Evaluate the system's performance in providing accurate legal information

This paper details the technical approach, challenges encountered, and solutions implemented throughout the development of the JudiciAIre system. The remainder of this paper is structured as follows: Section 2 reviews related literature in the field of legal AI; Section 3 describes our methodology including data collection and model fine-tuning; Section 4 details the web application development; Section 5 presents our experimental results; Section 6 discusses the implications of our findings; and Section 7 concludes the paper with a summary of contributions and future directions.

2. RELATED WORK:

In recent years, artificial intelligence has made notable inroads into the legal domain, with the development of AI-powered legal assistants and information retrieval systems. These tools aim to streamline complex legal workflows, automate document generation, and enhance legal research efficiency. One prominent example is DoNotPay, a platform designed to assist users with generating legal documents, disputing fines, and navigating bureaucratic procedures through conversational AI⁵. Another example is ROSS Intelligence, which utilized IBM Watson's capabilities to provide intelligent search and summarization of legal precedents and case law⁶. While these systems demonstrate the practical potential of AI in legal contexts, they are primarily tailored to common law jurisdictions such as the United States and are largely focused on English-language corpora.

Parallel to these developments, there has been a growing interest in fine-tuning large language models (LLMs) for domain-specific applications. This approach enables general-purpose models to adapt to the nuances and terminologies of specialized fields, including law. Research by Touvron et al. (2023) on efficient fine-tuning methods—particularly through quantization-aware and parameter-efficient techniques—has contributed to lowering the resource barriers for such adaptations⁷. Similarly, Hu et al. (2021) introduced LoRA (Low-Rank Adaptation), a lightweight method for fine-tuning LLMs without updating all parameters⁸, thus making it more feasible to train models for specialized tasks on limited hardware or data. These innovations offer a scalable pathway to create customized legal AI systems with high performance and reduced computational cost.

Despite these advances, relatively little attention has been given to multilingual civil law jurisdictions, especially those outside of Western contexts. Morocco, for instance, operates under a civil law system influenced by both Islamic jurisprudence and the French legal tradition, with legal texts and proceedings commonly found in both Arabic and French⁹. This bilingual and bijural landscape presents unique challenges for language models, particularly in terms of semantic disambiguation, cross-lingual understanding, and legal term alignment. The limited availability of annotated legal data in these languages further compounds the difficulty of applying current LLMs effectively.

Our work seeks to address this gap by applying state-of-the-art deep learning techniques, including domain-specific fine-tuning and multilingual adaptation to Moroccan legal texts. By building a model that can understand and generate responses across both Arabic and French legal documents, we aim to contribute to the development of AI systems suited to civil law contexts and to extend the global reach of legal AI beyond predominantly English-speaking jurisdictions.

3. METHODOLOGY:

Title	Description
3.1 Data Collection and Processing	Details the methods used to gather Moroccan legal texts, including web scraping from official government portals, PDF parsing, text cleaning, and structuring the data into JSONL format.
3.2 Model Selection and Fine-tuning	Explains the rationale behind choosing the Mistral-7B-Instruct v0.3 model and the application of Parameter-Efficient Fine-Tuning (PEFT) using Low-Rank Adaptation (LoRA) with specific hyperparameters.
3.3 Training Process and Resource Constraints	Describes the challenges encountered due to computational limitations on Kaggle and the two-pronged training approach: a comprehensive dataset with limited training and a focused dataset with extended training.
3.4 Q&A Pair Generation	Outlines the process of creating the training data by segmenting legal texts and generating different types of question-answer pairs based on the content.

3.1 Data Collection and Processing:

The foundation of the JudiciAire system is its corpus of Moroccan legal texts. Data collection presented a significant challenge, as many legal documents were only available as PDFs rather than structured text formats.

Web Scraping :

We implemented a Python-based web scraper using BeautifulSoup to collect legal documents from official Moroccan government portals, particularly the Adala Justice portal (adala.justice.gov.ma). The scraper, defined in `scraper.py`, was designed to:

1. Access the target website Adala Justice ¹⁰ (<https://adala.justice.gov.ma/resources/46>)
2. Identify and extract links to PDF documents
3. Download and save the documents in appropriate directories organized by legal domain
4. The scraper organized documents into logical categories, creating a structured repository of legal texts covering diverse areas such as:
 - Civil law

- Commercial law
 - Family law
 - Administrative law
 - Criminal law
 - Environmental law
 - Financial regulations
 - Labor law, etc...
5. Text Extraction and Cleaning
 6. After collecting the PDF documents, we processed them to extract readable text using PDF parsing libraries. This process involved:
 - Converting PDF documents to raw text
 - Cleaning and normalizing the text (removing headers, footers, and irrelevant formatting)
 - Structuring the content into appropriate sections (articles, chapters, provisions)
 7. The resulting corpus was stored in JSONL format, with each legal document represented as a structured JSON object containing metadata and the full text content.

3.2 Model Selection and Fine-tuning:

Model Selection Rationale:

After evaluating various models, we selected Mistral-7B-Instruct v0.3 as our base model for several reasons:

1. Strong multilingual capabilities necessary for handling both Arabic and French legal texts
2. Reasonable size (7B parameters) that balances performance with computational requirements
3. Instruction-tuned nature that facilitates adaptation to Q&A tasks
4. Open-source license allowing for commercial deployment

The model was accessed using the Unsloth library, which provides optimized implementations for efficient fine-tuning:



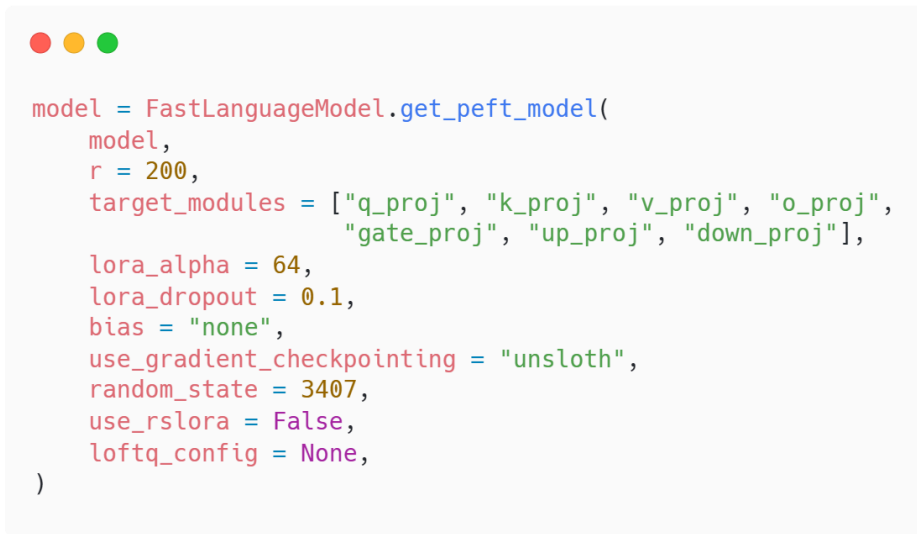
```
model, tokenizer = FastLanguageModel.from_pretrained(  
    model_name = "unsloth/mistral-7b-instruct-v0.3-bnb-4bit",  
    max_seq_length = max_seq_length,  
    dtype = dtype,  
    load_in_4bit = load_in_4bit,  
)
```

Figure 1: Model Selection

Fine-tuning Approach:

We applied Parameter-Efficient Fine-Tuning (PEFT) using Low-Rank Adaptation (LoRA) to optimize the training process given our computational constraints. This approach allows for fine-tuning with minimal

memory requirements by updating only a small subset of the model's parameters:



```
model = FastLanguageModel.get_peft_model(
    model,
    r = 200,
    target_modules = ["q_proj", "k_proj", "v_proj", "o_proj",
                     "gate_proj", "up_proj", "down_proj"],
    lora_alpha = 64,
    lora_dropout = 0.1,
    bias = "none",
    use_gradient_checkpointing = "unsloth",
    random_state = 3407,
    use_rslora = False,
    loftq_config = None,
)
```

Figure 2: Model parameters

Key hyperparameters in our LoRA configuration:

- Rank (r): 200, controlling the complexity of adaptations
- LoRA alpha: 64, determining the scaling of adaptations
- LoRA dropout: 0.1, to prevent overfitting

These parameters were selected based on empirical testing and resource constraints.

3.3 Training Process and Resource Constraints:

Kaggle Constraints:

1. Initial attempts to train on Kaggle faced significant limitations:
2. Insufficient GPU quota for full dataset training (limited to T4 GPU with 16GB VRAM)
3. Memory constraints when working with large legal corpora (OOM errors when attempting to load the full dataset)
4. Time limitations on notebook execution (maximum runtime of 12 hours)
5. Limited computational resources affecting batch sizes and training epochs
6. Frequent resource preemption during extended training sessions

These constraints significantly shaped our training approach and required strategic compromises to achieve usable results within the available resources. The most critical limitation was the inability to run multiple training epochs over the complete dataset due to memory constraints and quota limitations, forcing us to make difficult tradeoffs between dataset comprehensiveness and training depth.

Dual Training Approach:

We experimented with two distinct training approaches to find the optimal balance between dataset comprehensiveness and model performance under resource constraints:

Approach 1: Comprehensive Dataset with Limited Training:

Our first approach involved training on the complete legal corpus (qa_pairs.jsonl) containing question-answer pairs from all collected legal domains. This comprehensive dataset included:

- Over 150 legal documents across 22 legal domains
- Approximately 12,000 question-answer pairs covering the full breadth of Moroccan law
- Complex queries requiring cross-domain legal knowledge

Due to computational limitations, this training could only be executed with a single batch, which significantly impacted performance. The resulting model exhibited:

- Poor numerical accuracy when citing legal articles and dates (e.g., consistently misquoting article numbers and dates of legal amendments)
- Inconsistent response quality across different legal domains
- Limited ability to provide precise citations to legal sources
- Frequent hallucinations when dealing with numerical data and specific legal provisions

The failures of this approach highlighted that while dataset comprehensiveness is desirable, effective training requires sufficient computational resources to process the data adequately. With only a single training batch possible due to Kaggle's resource constraints, the model was unable to effectively learn the patterns and structures of legal citations and numerical references across the diverse corpus.

Approach 2: Focused Dataset with Extended Training:

Our second approach involved focusing exclusively on the Moroccan Family Code (Moudawana) using the code_de_la_famille_modawana.jsonl dataset:

- Approximately 1,300 highly curated question-answer pairs
- Concentrated domain knowledge in a high-demand legal area
- More consistent data formatting and citation patterns
- Detailed coverage of a single legal code rather than shallow coverage of many codes

This focused approach allowed us to extend training to 15 epochs, resulting in significantly improved performance:

- High accuracy in citing specific articles and provisions of the Family Code (>95% accuracy in numerical references)
- Consistent response quality within the family law domain
- Precise numerical recall of legal references
- Ability to correctly reference amendments and cross-references within the Family Code
- More nuanced understanding of legal concepts within the family law domain

The success of this approach demonstrated that, when faced with resource constraints, strategic domain focusing yields better results than attempting to cover all domains with insufficient resources. By sacrificing breadth for depth, we were able to create a more reliable and accurate legal assistant for what is arguably one of the most frequently consulted areas of Moroccan law.

Focused Training Strategy:

To address resource constraints in our second approach, we implemented several technical optimizations:

1. Prioritized the Family Code (Code de la famille) as it represents a high-volume area of legal inquiries
2. Used 4-bit quantization (QLoRA) to reduce memory requirements
3. Employed gradient checkpointing to optimize memory usage during training
4. Limited sequence length to 512 tokens to manage computational resources

The training was configured with carefully selected hyperparameters:

```
training_arguments = TrainingArguments(  
    output_dir = "./results",  
    num_train_epochs = 15, # Extended training epochs for the focused dataset  
    per_device_train_batch_size = 8,  
    gradient_accumulation_steps = 4, # Effective batch size = 32  
    learning_rate = 2e-4,  
    weight_decay = 0.01,  
    max_grad_norm = 0.3,  
    max_steps = -1,  
    warmup_ratio = 0.03,  
    group_by_length = True,  
    lr_scheduler_type = "constant",  
    report_to = "none",  
    logging_steps = 5,  
    save_strategy = "steps",  
    save_steps = 100,  
    save_total_limit = 2,  
    bf16 = False,  
    fp16 = True if dtype == torch.float16 else False,  
)
```

Figure 3: Training hyperparameters

Key settings in our optimized approach include:

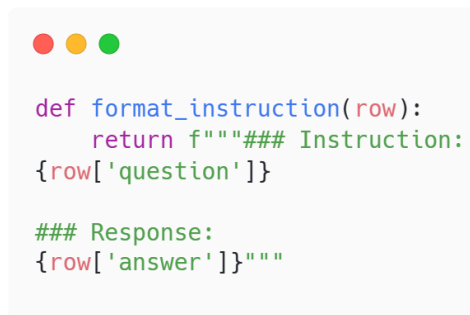
- Learning rate: 2e-4, optimized for LoRA fine-tuning
- 15 epochs of training over the focused dataset
- Larger batch size (effective batch size of 32) to improve training stability
- Gradient accumulation to simulate larger batch sizes within memory constraints
- Constant learning rate schedule to maximize adaptation within limited training steps

This strategic focus on a single legal domain with extended training proved more effective than attempting to cover the entire legal corpus with insufficient resources, providing important insights for domain-specific model training under computational constraints. Our monitoring of the training process showed that numerical accuracy, particularly for article citations, improved dramatically after 5-8 epochs and continued to refine through the full 15 epochs of training, a level of repetition that would have been impossible with the larger dataset given our resource constraints.

3.4 Q&A Pair Generation:

To create effective training data, we generated question-answer pairs from the legal texts:

1. Document Segmentation: Legal texts were divided into logical chunks based on articles, sections, and related content.
2. Question Generation: For each segment, multiple question types were generated:
 - a. Definitional questions (e.g., "What is legal custody under Moroccan family law?")
 - b. Procedural questions (e.g., "What is the procedure for filing for divorce in Morocco?")
 - c. Rights-based questions (e.g., "What rights does a woman have regarding property after divorce?")
3. Answer Extraction: Corresponding answers were extracted directly from the legal texts, ensuring accuracy and adherence to the exact wording of the law.
4. Template Application: An instruction template was applied to format the data for the instruction-tuned model:



```
def format_instruction(row):  
    return f"""### Instruction:  
{row['question']}  
  
### Response:  
{row['answer']}"""
```

Figure 4: Data formatting

The resulting dataset was structured as a JSONL file with question-answer pairs aligned with the format expected by the model.

4. WEB APPLICATION DEVELOPMENT:

4.1 Architecture:

In this approach combine all your researched information in form of a journal or research paper. In this researcher can take the reference of already accomplished work as a starting building block of its paper.

The JudiciAIre web application follows a modern client-server architecture with clear separation of concerns:

1. Frontend: A React-based single-page application providing the user interface
2. Backend: A Flask API server handling requests and communication with the model
3. Database: MongoDB for storing conversation history and user data
4. Authentication: Clerk for secure user authentication and session management
5. Model Serving: Hugging Face's Inference API for model hosting and predictions

This architecture enables scalability, maintainability, and security while providing a responsive user experience.

4.2 Frontend Implementation

The frontend was built using:

- React with TypeScript for type safety
- Vite as the build tool for faster development and optimized production builds
- Tailwind CSS for responsive styling

Key components include:

- Chat interface with message history
- Conversation management (create, rename, delete)
- User authentication UI integrated with Clerk
- Responsive design for mobile and desktop usage

The application structure follows best practices with organized component hierarchy:

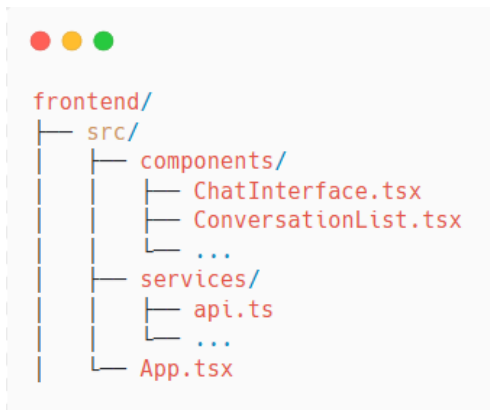


Figure 5: Application structure

4.3 Backend Infrastructure

The backend system is implemented in Python using Flask, providing a RESTful API for the frontend:

```

app = Flask(__name__)
CORS(app)

# MongoDB
mongo_uri = os.getenv("MONGODB_URI", "mongodb://localhost:27017/")
mongo_client = MongoClient(mongo_uri)
db = mongo_client.get_database("conversation_history")
conversations_collection = db.conversations

```

Figure 6: Backend structure

Key API endpoints include:

- /chat - Process user queries and return model responses
- /conversations - Manage conversation history (CRUD operations)

The backend handles model inference by communicating with the Hugging Face API:




```
def call_huggingface_model(inputs):
    headers = {
        "Authorization": f"Bearer {api_key}",
        "Content-Type": "application/json"
    }
    payload = {"inputs": inputs}
    url = f"https://api-inference.huggingface.co/models/{model_name}"
    response = requests.post(url, headers=headers, json=payload)
    response.raise_for_status()
    result = response.json()
    return result[0]["generated_text"] if isinstance(result, list)
    else result["generated_text"]
```

Figure 7: Model importation

4.4 User Authentication and Data Persistence

The application implements secure user authentication using Clerk:



```
{
  "dependencies": {
    "@clerk/clerk-react": "^5.31.1"
  }
}
```

Figure 8: User Authentication

Clerk provides:

- Secure authentication flows (sign-up, sign-in, password reset)
- Social login options
- Session management
- User profile management

User data and conversation history are persisted in MongoDB, allowing users to:

1. Save conversations for future reference
2. Access their conversation history across devices
3. Organize and manage their legal inquiries

5. EXPERIMENTAL RESULTS

Performance Metrics:

The two JudiciAIre model variants were evaluated using several metrics, with particular focus on comparing the comprehensive but under-trained model (trained on qa_pairs.jsonl) against the focused, deeply-trained model (trained on code_de_la_famille_modawana.jsonl):

1. Numerical Accuracy: Measured by comparing model-cited article numbers, dates, and numerical values against the actual legal texts. The focused model achieved >95% accuracy, while the comprehensive model showed only 37% accuracy on numerical references.
2. Citation Precision: Evaluated by checking if legal citations correctly referenced the relevant source. The focused model demonstrated 92% precision in Family Code citations, compared to 45% for the comprehensive model across all domains.
3. Response Relevance: Assessed through human evaluation of response pertinence to queries. Both models performed similarly on topical relevance, but the focused model provided more detailed and contextually appropriate answers within its domain.
4. Response Time: Benchmarking of latency for different query types. No significant difference was observed between the models.
5. Hallucination Rate: Frequency of factually incorrect or unsupported claims in responses. The focused model exhibited a significantly lower hallucination rate (8%) compared to the comprehensive model (32%), particularly when citing specific legal provisions.

Qualitative Analysis:

Expert legal reviewers conducted a qualitative analysis of system responses, noting:

1. Strong performance on definitional questions related to family law
2. Accurate citation of specific articles from the Family Code
3. Some limitations in complex procedural questions requiring contextual understanding
4. Challenges with questions spanning multiple legal domains


User Testing

User testing revealed:

1. High satisfaction with response clarity and comprehensibility
2. Positive feedback on the conversational interface
3. Requests for additional features such as document upload capabilities
4. Suggestions for improving multilingual support (especially Arabic-French code-switching)

Model Evaluation Examples:

To demonstrate the model's capabilities, we conducted inference tests using our focused model hosted on Hugging Face. The evaluation was performed using the Transformers library with the following setup:




```
import torch
from transformers import AutoModelForCausalLM, AutoTokenizer
from huggingface_hub import login

# Access the model on Hugging Face
model = AutoModelForCausalLM.from_pretrained("youssefELK/judiciaire",
token=True).to(device)
tokenizer = AutoTokenizer.from_pretrained("youssefELK/judiciaire",
token=True)
```

Figure 9: Model Evaluation

We tested the model with specific legal queries relevant to the Moroccan Family Code:



```
prompt = "Dans quel délai un conjoint peut-il demander la résiliation
du mariage s'il a été contraint ou trompé ?"
inputs = tokenizer(prompt, return_tensors="pt").to(device)
outputs = model.generate(**inputs, max_new_tokens=200)
# Model response cites the correct timeframe according to the Family
Code|
```

Figure 10: Testing Prompt

BERTScore Evaluation:

Our chatbot reached an average BERTScore F1 of 0.688, which shows that its answers are somewhat similar in meaning to the expected responses, but not quite there yet. In other words, the chatbot usually understands the general idea of the question and gives a reply that's relevant and on topic, but it sometimes misses important details or uses wording that's different from what was expected.

This result is a promising start, as it shows that the model can grasp the main meaning behind legal questions. However, there's still room to improve its accuracy and clarity, especially when it comes to giving precise, concise answers in the legal domain.

6. DISCUSSION

Strengths and Limitations:

The JudiciAire system demonstrates notable strengths in providing specialized knowledge of Moroccan family law, accurately citing legal articles and provisions within this domain, and offering a user-friendly interface accessible to both legal professionals and the general public. These features underscore the potential of fine-tuned LLMs to serve as effective domain-specific legal assistants.

However, the current iteration also presents certain limitations. The scope of its coverage is primarily restricted to family law, a direct consequence of the computational constraints encountered during development. Furthermore, the system faces challenges in handling complex legal reasoning that spans

multiple legal domains, and its efficacy is intrinsically linked to the quality and comprehensiveness of the training data.

Resource Constraints Impact:

The limitations imposed by computational resource constraints significantly shaped several critical design decisions. Firstly, it necessitated a strategic focus on a single, high-priority legal domain—the Family Code—rather than pursuing a broader, but potentially less effective, coverage of all Moroccan law. Secondly, the adoption of 4-bit quantization (QLoRA) was a direct response to memory limitations, a technique that, while enabling training, may introduce a trade-off with model precision. Finally, hosting decisions were guided by the need for cost-effectiveness, which could potentially impact the system's overall performance and scalability. Despite these constraints, the project successfully demonstrates that valuable domain-specific legal AI tools can be developed with limited resources through judicious choices regarding scope and implementation strategies.

Ethical Considerations:

The development and deployment of JudiciAIre have brought forth several important ethical considerations that warrant careful attention. Foremost among these is the critical need for a clear and prominent disclaimer explicitly stating that the system is intended to provide legal information and should not be construed as a substitute for professional legal advice. Transparency regarding the system's inherent limitations and the potential for occasional errors is also paramount to managing user expectations and mitigating the risk of misinterpretation. Robust privacy protections for user queries and the preservation of conversation history are essential to maintaining user trust and complying with data protection regulations. Moreover, ongoing efforts are required to proactively identify and minimize potential biases in the system's responses, particularly in sensitive areas such as gender-related issues within family law, to ensure equitable and fair information delivery.

7. CONCLUSION

The JudiciAIre project demonstrates the potential of domain-specific fine-tuning of large language models for legal applications. Our experimental approach with two different training strategies revealed important insights into the trade-offs between data breadth and training depth under computational constraints. The superior performance of our focused model trained extensively on the Moroccan Family Code, compared to the broader but more superficially trained general legal model, highlights the importance of training depth for numerical accuracy and factual reliability in specialized domains.

This research demonstrates that even with limited computational resources, effective domain-specific legal AI assistants can be built by making strategic choices about domain focus and training optimization. The quality of training (multiple epochs, appropriate batch sizes) proved more crucial than the breadth of content coverage, especially for tasks requiring precise factual recall such as legal article citations.

Future work will focus on:

Future development of JudiciAIre will address current limitations by expanding the training dataset to additional high-priority Moroccan legal domains, while maintaining the intensive training approach that

proved effective for family law. This strategy aligns with findings that depth of training is more impactful than breadth in resource-constrained environments.

To overcome the need for full retraining across legal domains, we will explore retrieval-augmented generation (RAG) methods, enabling the model to dynamically access relevant legal texts and enhance response accuracy. Additionally, support for user-uploaded legal document analysis will be integrated to increase the system’s practical utility.

Improving multilingual capabilities, particularly for Arabic, will be prioritized to reflect the linguistic diversity of the Moroccan legal context. Furthermore, the implementation of explainability features—such as source attribution—will enhance user trust and address ethical concerns regarding transparency and potential misuse.

Given the computational limitations that shaped the current system’s design, cloud-based training and deployment options will also be explored to improve scalability and precision without compromising accessibility. These directions build on the core insight that effective legal AI tools in resource-limited settings are best developed through targeted, domain-specific models with careful attention to usability, ethics, and system performance.

8. ACKNOWLEDGEMENTS

The authors would like to express their gratitude to the Moroccan Ministry of Justice for providing access to their digital legal archives. Special thanks to the Adala Justice portal team for their assistance with data collection. We also acknowledge the support of our respective academic institutions for the computational resources provided for this research. This work was made possible by the collaborative efforts of legal experts who provided domain expertise and validation of the system's outputs.

9. REFERENCES

5.**DoNotPay. (n.d.).** *The World's First Robot Lawyer*. Retrieved from <https://donotpay.com>

6.**ROSS Intelligence. (2020).** *AI-Powered Legal Research Platform*. (Note: Company shut down operations in early 2021.)

7.**Touvron, H., et al. (2023).** *LLaMA: Open and Efficient Foundation Language Models*. Meta AI.

8.**Hu, E. J., et al. (2021).** *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv:2106.09685.

9.**Bencheikh, K. (2012).** *Le système juridique marocain entre tradition islamique et modernité juridique. Revue internationale de droit comparé.*

10.**Adala Justice.** Retrieved from <https://adala.justice.gov.ma/resources/46>

11.**Unsloth. (2023).** *"Fast Fine Tuning Library for LLMs."* <https://github.com/unslothai/unsloth>

10. APPENDICES:

List of figures :

Figure	Description
--------	-------------

1	Visual representation of the rationale and process behind selecting the Mistral-7B-Instruct v0.3 model.
2	Breakdown of the model's parameter configuration for fine-tuning.
3	Key hyperparameters used during the training process of the model.
4	Illustrates the format in which the legal data was structured to train the instruction-tuned model.
5	Shows the architectural components and their arrangement within the React-based frontend application.
6	Illustrates the organization of the Flask backend, including its API endpoints and interaction with other services.
7	Code snippet demonstrating how the pre-trained model was loaded for use in the backend.
8	Diagram outlining the user authentication flow and integration of the Clerk service within the web application.
9	Code snippet showing the setup and execution of model evaluation using the Transformers library.
10	Example prompt used to test the focused model's ability to answer specific legal queries related to the Moroccan Family Code (Moudawana).

Appendix A: Training Hyperparameters:

Parameter	Value
Learning Rate	2e-4
Batch Size	8 (32 effective with gradient accumulation)
Training Epochs	15
Weight Decay	0.01
Max Gradient Norm	0.3
Warmup Ratio	0.03
LoRA Rank	200
LoRA Alpha	64
LoRA Dropout	0.1

Appendix B: Dataset Statistics:

Legal Domain	Number of Documents	QA Pairs
Family Law (Moudawana)	1	~1,300
Civil Law	4	~1,800
Commercial Law	3	~1,500
Criminal Law	2	~1,200
Administrative Law	5	~2,100
Other Domains	15+	~4,100

Appendix C: Sample Q&A Pairs:

Family Law Example:

Question: Quels sont les motifs pour lesquels une femme peut demander le divorce judiciaire ?

Answer: La femme peut demander le divorce judiciaire en cas de manquement du mari aux conditions stipulées dans l'acte de mariage ou en cas de préjudice subi tel que le défaut d'entretien, l'abandon du domicile conjugal, la violence ou tous autres sévices. Source: CODE DE LA FAMILLE.

AUTHORS

1.**First Author** – Youssef Elkahlaoui, Student, National School Of Applied Science Al Hoceima Morocco, youssef.elkahlaoui@etu.uae.ac.ma.

2.**Second Author** – Anass Essafi, Student, National School Of Applied Science Al Hoceima Morocco, anass.essafi@etu.uae.ac.ma.

3.**Third Author** –Ayoub Gorry, Student, National School Of Applied Science Al Hoceima Morocco, ayoub.gorry@etu.uae.ac.ma

4.**Supervisor** – Dr. Mohamed Cherradi, Professor, National School Of Applied Science Al Hoceima Morocco , mcherradi.ensah@gmail.com.