# Project Documentation - Legal Dataset Generation

Moisă Clementina-Ioana, Muntean Larisa-Maria, Sălejan Maria

## Synthetic Romanian Legal Corpus Generation

## 1 Methodology

The proposed approach utilizes a multi-stage pipeline designed to generate synthetic Romanian legal documents (specifically notary powers of attorney and declarations, requests, and contracts). The methodology circumvents the need for computationally expensive fine-tuning on a small dataset (approx. 40 documents) by leveraging In-Context Learning (ICL) and a Retrieval-Augmented Generation (RAG) inspired workflow. The pipeline consists of four distinct phases:

### 1.1 Data Preparation and Cleaning

The initial dataset consisted of 40 raw, anonymized real-world templates for notary powers of attorney and declarations, containing placeholders (e.g., "..."). An initial Large Language Model (LLM), **RoLlama3-8b-Instruct-DPO-2025-04-23**, pass was employed to "repair" these documents, replacing ellipses with coherent synthetic entities using a probabilistic approach. This resulted in a "Golden Standard" library of clean, complete text files stored in a persistent environment (Google Drive).
The dataset also included around 100 synthetic court requests, generated based on standard templates in JSON format. These were anonymized, respecting good ethical practices and GDPR.
Moreover, a total of 121 PDF contracts were converted into structured question-answer JSON format using a 'Generate [Contract Title]' prompt pattern.

### 1.2 Scenario Generation (AI Screenwriter)

To ensure corpus diversity and prevent overfitting to specific templates, a distinct LLM instance of **RoLlama3-8b-Instruct-DPO-2025-04-23** acts as a "Screenwriter." Instead of using static procedural generation, this agent generates complex JSON metadata containing diverse legal scenarios (e.g., real estate sales, vehicle registration, travel declarations for minors, succession). This introduces semantic variety and realistic edge cases into the input data.

### 1.3 Context-Aware Generation

The generation phase employs a retrieval mechanism. Based on the metadata generated in the previous step, the system retrieves the most semantically relevant template from the "Golden Standard" library. The LLM is then prompted using a One-Shot strategy, utilizing specific prefix patterns to trigger the generation:

Prompt pattern used for requests generation

```
Redacteaza o cerere juridica completa in limba romana, pe baza urmatoarelor
   elemente:
- TipCerere: {tip} (ex: chemare in judecata, apel, recurs)
- Parti implicate: {parti} (persoane fizice sau juridice)
- Context/Scop: {scop} (ex: reglementare, dovada, solutionare conflict, constatare
   fapt)
- Conflict juridic: {conflict} (doar pentru cereri de chemare in judecata)
- Argumente si probe: {argumente}
- Termen dorit: {termen}
- Ton: {ton}
- Lungime aproximativa: {lungime} cuvinte
```

```
Structura cererii trebuie sa includa:
- Titlu
- Preambul
- Corpul documentului (cu articole/paragrafe numerotate)
- Semnaturi
- Data
- Stampile (daca este cazul)

Asigura respectarea cerintelor legale si a formalitatilor prevazute de lege.
```

During this process, the model is provided with the retrieved template as a strict stylistic guide and instructed to rewrite it using the new synthetic metadata. This ensures 100% adherence to legal formatting and terminology while varying the factual content, effectively transforming the raw metadata into fully-formed, legally-sound JSON question-answer pairs.

### 1.4 Unified Critic (Evaluation)

A final validation step is performed by a **Critic** agent, implemented using the **RoMistral 7B Instruct** model. This model is prompted as an *AUDITOR NOTARIAL EXPERT*, instructed to review generated legal documents according to a structured evaluation schema.

**Prompt:** The model is asked to analyze each document and evaluate the following aspects:

- **Structural Integrity (structura):** Checks whether the document has a correct title, a logical structure (introduction, body, conclusion), and completeness (no missing or incomplete clauses).
- **Legal Language (limbaj):** Evaluates whether the text is formal, neutral, and uses correct legal terminology without colloquial expressions or ambiguities.
- **Purpose Compliance (respectare_scop):** Assesses whether the document clearly mentions the mandant, the mandatar, and explicitly fulfills the requested purpose (e.g., "reprezentarea intereselor în instanță").

**Evaluation Schema:** The model is instructed to return a **strict JSON** with four fields:

- `structura` (int 1–10) — structural quality score.
- `limbaj` (int 1–10) — legal language quality score.
- `respectare_scop` (int 1–10) — alignment with requested purpose.
- `observatii` — concise textual remarks.

**Results:** After processing all generated documents, the average scores obtained were:

- Average `structura`: 7.47
- Average `limbaj`: 8.47
- Average `respectare_scop`: 6.81
- **Overall mean**: 7.58

Only documents that satisfy structural, linguistic, and purpose criteria are included in the final JSONL dataset.

# 2 Model Specifications

The core model used for most agents (Cleaner, Screenwriter and Generator) is **RoLlama3-8b-Instruct-DPO-2025-04-23**, a Llama 3 derivative fine-tuned for the Romanian language, while the Critic role was given to **RoMistral 7B Instruct** model. To facilitate execution on consumer-grade hardware (Google Colab T4 GPU with 16GB VRAM), the model was loaded using 4-bit quantization (NF4) via the `bitsandbytes` library. This configuration allowed for an efficient inference pipeline without significant degradation in linguistic quality.
To ensure linguistic variety while maintaining legal rigor, the model was configured with a conservative **temperature of 0.4** for generating requests. Their output length was capped at **500 max new tokens** to favor concise, high-density legal drafting.

# 3 Results

The pipeline successfully generated a synthetically augmented corpus that mimics the stylistic nuances of the 27 original source documents. The implementation of the "AI Screenwriter" and template retrieval mechanism solved the issue of mode collapse, ensuring that the output covers a wide taxonomy of legal acts. The "Unified Critic" loop effectively filtered out hallucinations and incomplete generations, resulting in a high-quality JSONL dataset suitable for future downstream tasks such as Named Entity Recognition (NER) or legal text classification.
To quantitatively evaluate the fidelity of the generated requests, we calculated the **BLEU** metric, achieving a score of 0.407. This indicates a high degree of structural and lexical alignment with our curated 'Golden Standard' library. While BLEU scores above 0.40 are typically representative of high-quality synthesis, in the legal domain, this value confirms that the system successfully preserved formalistic integrity while integrating new synthetic metadata

# 4 Comparison with State of the Art

The landscape of Romanian legal NLP has evolved from static corpus collection to discriminative deep learning models. Our approach represents a shift towards generative, agentic workflows. This section compares our methodology with two landmark contributions in the field: the MARCELL corpus [1] and the jurBERT model [2].

## 4.1 Static Resource Collection: The MARCELL Project

Văduva et al. (2020) established a significant benchmark by creating a massive national corpus of Romanian legislative texts. Their objective was to create a linguistically processed resource comparable to other EU languages.

- **Methodology:** The approach relied on web crawling official portals to collect over 144,000 legislative documents (laws, decisions, regulations) published between 1881 and 2018.
- **Pipeline:** The processing utilized the RELATE portal and TEPROLIN platform. It employed traditional NLP tools such as TTL (for tokenization and lemmatization) and NLP-Cube (neural dependency parsing). Named Entity Recognition (NER) was performed using Conditional Random Fields (CRF), while terminology was annotated using EuroVoc descriptors.
- **Contrast with Our Approach:** While MARCELL provides an invaluable snapshot of *real* historical data, it is static and bound by data privacy constraints (real PII). Our pipeline, conversely, is dynamic. Instead of collecting existing documents, we synthesize an infinite number of realistic variations. Furthermore, our agentic "Critic" replaces the static CRF-based validation with semantic verification, allowing for the generation of data that maintains privacy by design.

## 4.2 Discriminative Understanding: jurBERT

Masala et al. (2021) advanced the field by moving from resource collection to semantic understanding, developing *jurBERT*, a domain-adapted model for legal judgement prediction.

- **Methodology:** The authors utilized a BERT-base architecture (Encoder-only) and applied a two-stage domain adaptation process. First, they performed continued pre-training on the MARCELL corpus and other judicial decisions to learn legal embeddings. Second, they fine-tuned the model for the specific task of predicting the outcome of lawsuits (admission or rejection).
- **Results:** jurBERT achieved state-of-the-art performance (F1 $\approx$ 86–88%), significantly outperforming multilingual BERT (mBERT) and general Romanian BERT (RoBERT).
- **Contrast with Our Approach:** jurBERT represents the pinnacle of *discriminative* AI in Romanian law—it excels at reading and classifying. Our RoLlama3-based approach utilizes *generative* AI (Decoder-only). While jurBERT requires massive computational resources for pre-training on millions of documents, our method utilizes In-Context Learning (ICL) and RAG to achieve high-fidelity drafting with minimal compute (T4 GPU). Our system is complementary to jurBERT: the synthetic data generated by our pipeline could theoretically be used to train future iterations of jurBERT-like models without privacy concerns.

### 4.3 Synthesis: Generative vs. Traditional Pipelines

The distinct advantages of the proposed Agentic Synthetic Data pipeline compared to the existing State of the Art are summarized in Table 1.

Table 1: Comparison of Methodologies in Romanian Legal NLP

| **Feature** | **MARCELL (2020)** | **jurBERT (2021)** | **Ours (Proposed)** |
|---|---|---|---|
| **Paradigm** | Resource Collection | Discriminative (Encoder) | Generative (Decoder/Agentic) |
| **Core Task** | Annotation & Storage | Classification & Prediction | Synthesis & Drafting |
| !**Data Source** | 144k Real Documents | Real Legal Corpora | 27 Templates + AI Screenwriter |
| **Privacy** | Requires Anonymization | Trained on Real Data | Private by Design (Synthetic PII) |
| **Validation** | Statistical (CRF/F1) | Accuracy/F1 Score | Semantic Agentic Critic |
| **Scalability** | Limited by availability | High compute cost | Infinite (Generation on demand) |

In conclusion, while MARCELL solved the problem of *data availability* and jurBERT solved the problem of *legal understanding*, our approach addresses the problem of *data scarcity and privacy* through agentic synthesis.

## 5 Application Integration

While the core of this research focuses on the generative capabilities of the model,this section outlines the transition from a standalone Large Language Model to a Python-based application.
The system generates documents in Romanian language, evaluates their quality automatically using an LLM-critic, and saves the generated documents and evaluations in JSON format.

In order to configure the requests, a template was established with the possible values of all parameters.

Template and Configuration

```
SYSTEM_MSG = "Esti un asistent care redacteaza cereri juridice clare..."

TEMPLATE = """
Redacteaza o cerere juridica completa in limba romana, pe baza urmatoarelor
    elemente:
- TipCerere: {tip}
- Parti implicate: {parti}
- Context/Scop: {scop}
- Conflict juridic: {conflict}
- Argumente si probe: {argumente}
- Termen dorit: {termen}
- Ton: {ton}
"""

types = ["chemare in judecata", "apel", "recurs"]
parti = ["persoana fizica", "persoana juridica"]
```

```
scopuri = ["reglementare", "dovada", "solutionare a unui conflict"]
conflict = ["exista conflict juridic", "nu este cazul unui conflict juridic"]
argumente = ["prezint probele anexate", "invoc prevederile legale aplicabile"]
tonuri = ["formal", "strict formal"]
```

Values are then selected to complete the template used for the prompt generation. The resulting document is saved in JSON format and evaluated based on previously mentioned criteria.

The model is called in a deterministic manner, with a low GPU memory usage, based on the code below.

Model Call and Evaluation Saving

```
def call_model(system, prompt, meta, temperature=0.0,
max_new_tokens=300):
full_prompt = f"System:   {system}\nUser:   {prompt}\
nAssistant:"
inputs = tokenizer(full_prompt, return_tensors="pt")
inputs = {k: v.to(model.device) for k, v in inputs.
items()}
with torch.no_grad():
outputs = model.generate(
**inputs,
max_new_tokens=max_new_tokens,
temperature=temperature,
do_sample=False,
pad_token_id=tokenizer.eos_token_id
)
decoded = tokenizer.decode(outputs[0],
skip_special_tokens=True)
generated_only = decoded[len(full_prompt):].strip()
generated_only = generated_only.replace("\\n", "\n")
data = {
"Cerere": meta,
"Evaluare": generated_only
}
try:
with open("evaluare_cereri.json", "r", encoding=
"utf-8") as f:
existing = json.load(f)
if not isinstance(existing, list):
existing = []
except FileNotFoundError:
existing = []
existing.append(data)
with open("evaluare_cereri.json", "w", encoding="utf-8") as f:
json.dump(existing, f, ensure_ascii=False,
indent=4)
torch.cuda.empty_cache()
return generated_only
```

# 6 Conclusion

This project demonstrates that high-fidelity synthetic legal data can be generated for low-resource languages (like Romanian) without extensive fine-tuning. By separating the workflow into specialized agents—specifically decoupling the scenario invention from the drafting process—and utilizing rigorous self-verification steps, we achieved a robust system capable of scaling a small set of templates into a large, diverse dataset for machine learning applications.

## References


[1] Văduva, V., et al. (2020). *Building a Representative Romanian Corpus*. In Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020), pp. 3291–3296. https://aclanthology.org/2020.lrec-1.337.pdf

[2] Masala, M., et al. (2021). *jurBERT: A Romanian BERT Model for Legal Judgement Prediction*. In Proceedings of the Natural Legal Language Processing Workshop 2021, pp. 86–94. https://aclanthology.org/2021.nllp-1.8.pdf