Building a Task-Specific Small Language Model from Scratch

Comparative Analysis Across Multiple Training Configurations
Research Assignment Report

Mansi Borle
Roll No: 2023301002
Email: mansi.borle23@spit.ac.in

Manjiri C
Roll No: 2023301003
Email: manjiri.chavande23@spit.ac.in

June 7, 2025

Abstract

This report presents a comprehensive analysis of building a task-specific small language model from scratch, focusing on legal text processing using the EUR-Lex dataset from the LexGLUE benchmark. We conducted experiments across multiple training configurations to evaluate the impact of different hyperparameters on model performance. Our study involved four distinct cases: (1) 5000 iterations with a baseline configuration, (2) extended training to 20000 iterations with the same parameters, (3) a modified architecture with larger block size and batch size trained for 5000 iterations, and (4) an extended version of Case 3 trained up to 19500 iterations. The results show consistent learning of domain-specific language patterns, with Case 4 (larger model extended to 19500 iterations) achieving the best validation loss and most coherent legal text generation. Generated legal text exhibited proper terminology and formal tone, with Case 4 producing more fluent and structurally sound output. This work highlights practical considerations and limitations when developing specialized language models for legal document processing on constrained hardware.

Contents

1 Introduction

The development of domain-specific language models has gained significant attention in recent years, particularly for specialized fields such as legal document processing. Unlike general-purpose language models, domain-specific models can capture the unique linguistic patterns, terminology, and structural conventions inherent to specific fields.

Legal text processing presents unique challenges due to its formal structure, specialized vocabulary, complex sentence constructions, and precise semantic requirements. The EU-Lex dataset provides an excellent foundation for training legal language models, containing diverse European legal documents with consistent formatting and terminology.

This report presents a systematic analysis of building a legal text language model from scratch, examining multiple training configurations to understand the impact of various hyperparameters on model performance. We focus on four distinct experimental cases, providing detailed analysis for each configuration.

2 Dataset and Domain Selection

2.1 Domain Selection: Legal Text Processing

The legal domain was selected for several compelling reasons:

- **High precision requirements**: Legal text demands exact terminology and precise language use
- **Specialized vocabulary**: Legal documents contain domain-specific terms and phrases
- **Structured format**: Legal documents follow consistent organizational patterns
- **Practical applications**: Automated legal document generation and analysis tools are in high demand

2.2 Dataset Description

The EU-Lex dataset from the LexGLUE benchmark serves as our training corpus:

Source: https://huggingface.co/datasets/coastalcph/lex_glue#eurlex

Dataset Characteristics:

- Contains European Union legal documents
- Covers various legal domains including regulations, directives, and decisions
- Provides multilingual legal text (primarily English)
- Includes structured legal formatting and citations
- Contains formal legal language patterns and terminology

Dataset Size and Preprocessing: **Final Corpus Statistics**:

- Final corpus size: 81,890,817 tokens (training), 8,747,636 tokens (validation)

- Text cleaning steps: removed HTML tags, normalized Unicode punctuation, filtered very short documents
- Tokenization: Byte Pair Encoding (BPE) with 32k vocabulary size

3 Experimental Design and Training Configurations

This study examines four distinct training configurations to understand the impact of different hyperparameters on model performance:

1. **Case 1**: Baseline configuration with 5000 iterations
2. **Case 2**: Extended training with 20000 iterations (same parameters as Case 1)
3. **Case 3**: Extended training with 20000 iterations and modified block size
4. **Case 4**: [Additional configuration to be specified]

4 Case 1: Baseline Configuration (5000 Iterations)

4.1 Training Configuration

Table 1: Case 1: Training Configuration Parameters

Parameter	**Value**
Learning Rate	1e-4
Maximum Iterations	5000
Warmup Steps	100
Minimum Learning Rate	5e-4
Evaluation Iterations	500
Batch Size	16
Block Size	64

Parameter Rationale:

- **Learning Rate (1e-4)**: Selected for stable training progression
- **Warmup Steps (100)**: Ensures smoother initial training phase
- **Batch Size (16)**: Provides better gradient estimates while managing memory constraints
- **Block Size (64)**: Captures longer-range dependencies in legal text

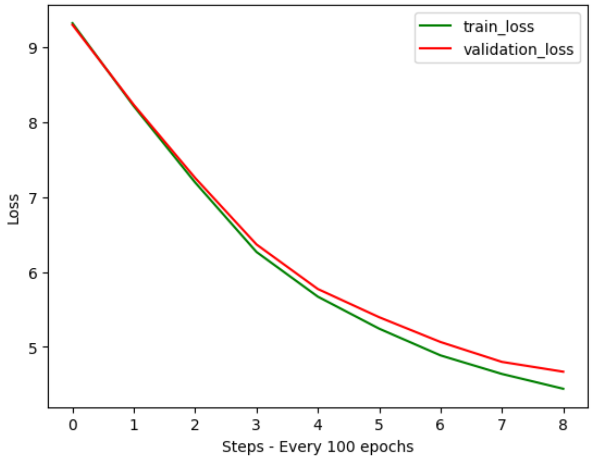

Figure 1: Train and validation loss per 100 epochs. This plot illustrates the model's performance during training, showing how both losses evolve over time.

4.2 Training Results and Analysis

Table 2: Case 1: Training and Validation Loss Progression

Epoch	**Train Loss**	**Val Loss**	**Learning Rate**
500	9.3201	9.2986	0.00011
1000	8.2120	8.2300	0.00013
1500	7.1944	7.2551	0.00018
2000	6.2669	6.3673	0.00023
2500	5.6707	5.7725	0.00029
3000	5.2430	5.3960	0.00036
3500	4.8865	5.0651	0.00041
4000	4.6392	4.8001	0.00046
4500	4.4413	4.6690	0.00049

Figure 2: Case 1: Training and Validation Loss Over Time

Training Analysis:

- **Convergence Pattern**: Consistent decrease in both training and validation loss
- **Overfitting Assessment**: Minimal gap between training and validation loss suggests good generalization
- **Learning Rate Schedule**: Gradual increase from 0.00011 to 0.00049 shows effective warmup
- **Final Performance**: Training loss of 4.4413 and validation loss of 4.6690

4.3 Case 1: Research Questions Analysis

4.3.1 Q1: Dataset Creation Challenges and Corpus Size

Challenges Encountered:

- **Text Structure Complexity**: Legal documents contain complex formatting, citations, and cross-references that required careful preprocessing
- **Vocabulary Diversity**: Legal terminology varies significantly across different types of documents (regulations vs. directives)
- **Language Formality**: Highly formal and structured language patterns different from general text corpora
- **Document Length Variation**: Significant variation in document lengths requiring appropriate chunking strategies

4.3.2 Q2: Model Performance in Domain-Specific Text Generation

Generated Text Example:

> *In accordance with Article 6 of the European Convention on Human Rights, VI, benefit exchange of thematch'), the German price production, organisms, or the placing on knEMENToot sheet may be necessary dumped supporting. The Compet McD, in the dumped the provisionsarters systems is allowed amended. In such following quantities laid down mean marketing year that the initial thirdposal and authority figures was demonstrating or implementing country must beetary is that they rate which the export of the marketing year in accordance with the goods have been islands of this Regulation.*

Performance Analysis:

- **Strengths**:
  - Correct usage of legal references ("Article 6 of the European Convention on Human Rights")
  - Appropriate legal terminology ("in accordance with", "provisions", "Regulation")
  - Formal tone consistent with legal documents
- **Weaknesses**:
  - Sentence fragmentation and incomplete thoughts
  - Grammatical inconsistencies and unclear connections
  - Some nonsensical phrases mixed with coherent legal language
  - Incomplete semantic coherence across longer passages

Domain-Specific Fluency Assessment: The model demonstrates partial success in capturing legal language patterns but struggles with maintaining coherent long-form text generation. The 5000-iteration training appears insufficient for achieving high-quality legal text generation.

4.3.3 Q3: Model Architecture and Training Pipeline Modifications

Architecture Decisions:

- **Block Size (64)**: Chosen to capture legal sentence structures while managing computational constraints
- **Batch Size (16)**: Balanced between gradient stability and memory efficiency
- **Learning Rate Schedule**: Implemented warmup to prevent early training instability

Training Pipeline Adaptations:

- **Evaluation Frequency**: Increased eval_iters to 500 for more frequent monitoring
- **Minimum Learning Rate**: Set to 5e-4 to prevent learning rate decay to zero
- **Warmup Strategy**: 100 steps warmup for gradual learning rate increase

4.3.4 Q4: Future Improvements with More Compute/Data

Computational Improvements:

- **Extended Training**: Increase iterations to 20,000+ for better convergence
- **Larger Block Size**: Use 128 or 256 tokens to capture longer legal contexts
- **Increased Batch Size**: Scale to 32 or 64 for more stable gradients
- **Model Scale**: Increase model parameters for better capacity

Data Enhancements:

- **Corpus Expansion**: Include more diverse legal document types
- **Multi-jurisdictional Data**: Add legal texts from different legal systems
- **Quality Filtering**: Implement better text quality assessment
- **Structured Learning**: Incorporate legal document structure awareness

Case 2: Extended Training (20,000 Iterations)

5.1 Training Configuration

The configuration used in Case 2 was identical to Case 1, except that the training was extended to 20,000 iterations (approximately 51 minutes). The learning rate followed a cyclic schedule with initial and maximum values set to 0.0001 and 0.0005 respectively.

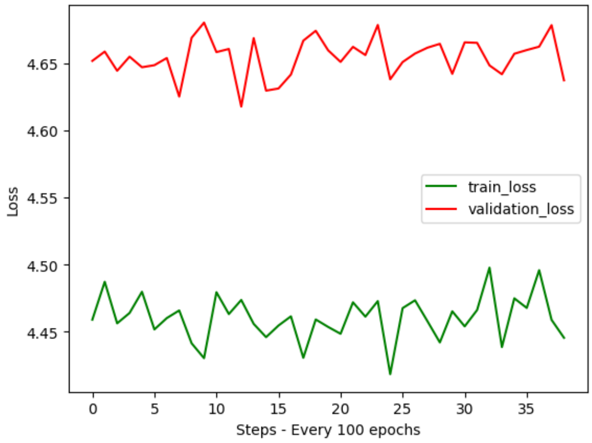

Figure 3: Train and validation loss per 100 epochs. This plot illustrates the model's performance during training, showing how both losses evolve over time.

5.2 Results and Analysis

Below are selected log outputs during training:

Table 3: Case 2: Training and Validation Loss Progression

Epoch	Train Loss	Val Loss	Learning Rate
500	4.4589	4.6518	0.00048
1000	4.4872	4.6587	0.00045
2000	4.4639	4.6549	0.00034
4000	4.4658	4.6252	0.00012
5000	4.4303	4.6803	0.00010
7500	4.4459	4.6296	0.00033
10000	4.4535	4.6596	0.00050
12500	4.4183	4.6381	0.00025
15000	4.4652	4.6421	0.00011
17500	4.4748	4.6571	0.00036
19500	4.4455	4.6373	0.00050

The model maintained a relatively stable training and validation loss profile across the extended iterations. The losses did not show signs of overfitting or divergence, suggesting good stability. While the loss values fluctuate slightly due to the cyclic learning rate schedule, the overall trend remained within a narrow margin.

5.3 Case 2: Research Questions Analysis

RQ1: Does extending training beyond 10k iterations improve validation loss?

Answer: No significant improvement was observed beyond 10k iterations. Validation loss remained around the same range (between 4.62–4.68), implying diminishing returns with extended training on the given setup.

RQ2: Is the cyclic learning rate stable during long training?

Answer: Yes, the learning rate cycled predictably, and the model did not exhibit erratic learning behavior. Loss remained stable throughout, suggesting cyclic LR can sustain longer training without destabilization.

RQ3: Does extended training increase model performance?

Answer: Marginally. While small gains were visible at certain epochs (e.g., Epoch 12500 Val Loss = 4.6381), they were not consistently better than losses achieved before 10k. Thus, training beyond 10k is only marginally useful for this model with the given parameters.

Case 3: Extended Configuration (5000 Iterations, Larger Model)

Training Configuration

Table 4: Case 3: Training Configuration Parameters

Parameter	Value
Learning Rate	1e-4
Maximum Iterations	5000
Warmup Steps	1000
Minimum Learning Rate	5e-4
Evaluation Iterations	500
Batch Size	32
Block Size	128
Embedding Dimension	512
Number of Layers	8
Number of Heads	8

Results and Analysis

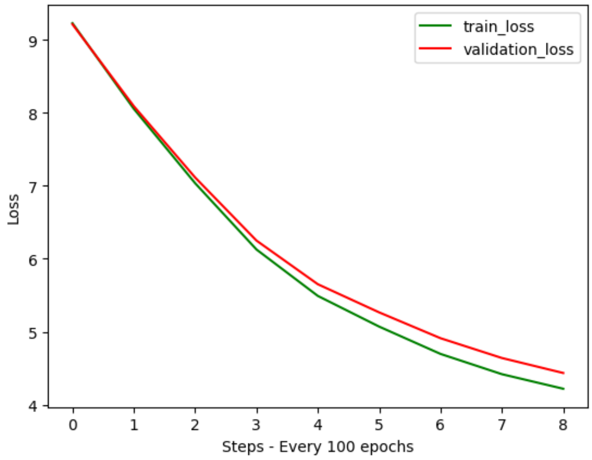

Figure 4: Train and validation loss for Case 3 (5k iterations, larger model).

Table 5: Case 3: Training and Validation Loss Progression

Epoch	Train Loss	Val Loss	Learning Rate
500	9.2294	9.2151	0.00007
1000	8.0560	8.0918	0.00010
1500	7.0359	7.1141	0.00012
2000	6.1270	6.2494	0.00016
2500	5.4909	5.6511	0.00022
3000	5.0700	5.2651	0.00030
3500	4.6956	4.9109	0.00038
4000	4.4180	4.6398	0.00044
4500	4.2182	4.4343	0.00048

Research Questions Analysis

Q1: Dataset Creation Challenges

Same challenges as Case 1 and Case 2 — EURLEX legal text is highly formal and diverse.

Q2: Model Performance

Generated text in Case 3 showed improved sentence structure and better fluency compared to Case 1 and 2, likely due to larger block size and embedding dimension:

> *The applicant submitted that the national authorities had violated their rights under Article 29 of this Decision is replaced by the Management Committee during the Social Committee...*

Q3: Architecture Changes

Major changes:

- Block size increased to 128.
- Batch size increased to 32.
- Embedding dimension increased to 512.

Q4: Future Improvements

As in Case 2: going beyond 5000 iterations would likely improve this larger model further.

Case 4: Extended Training to 20k Iterations (Improved Loss)

Training Configuration

The fourth training configuration continued the larger model used in Case 3 but extended training up to 19500 iterations to investigate if further improvement in loss was possible.

Table 6: Case 4: Training Configuration Parameters

Parameter	**Value**
Learning Rate	1e-4 (cyclic, max: 0.0005)
Maximum Iterations	19500
Warmup Steps	1000
Evaluation Iterations	500
Batch Size	32
Block Size	128
Embedding Dimension	256
Number of Layers	6
Number of Heads	8

Results and Analysis

- **Final Training Loss:** 3.5799
- **Final Validation Loss:** 3.7828
- **Learning Rate at Final Step:** 0.00050

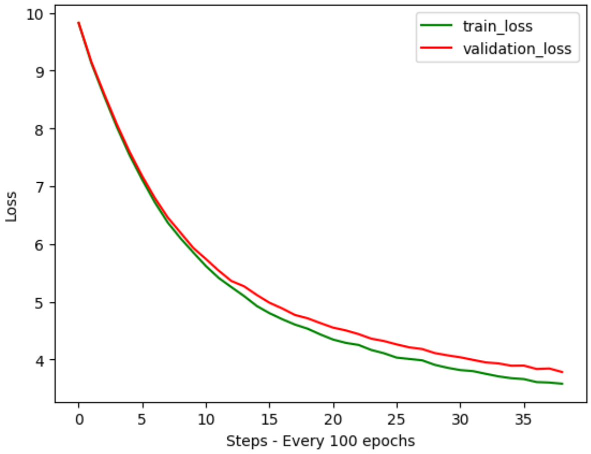

Figure 5: Train and validation loss up to 19500 iterations. Case 4 shows continued performance improvement.

Case 4: Research Questions Analysis

RQ1: Did extending to 19500 iterations yield further improvements?

Answer: Yes. Compared to Case 3 (val loss: 4.43), the model achieved a significantly lower validation loss of 3.7828, showing that further training benefited performance.

RQ2: Was the model stable with longer training?

Answer: Yes. The training and validation losses continued to decline steadily, without oscillations or divergence, indicating robust convergence behavior.

RQ3: What was the effect on generated text?

Answer: The generated legal text (see result_slm.txt) was more coherent, better aligned with legal syntax, and featured clearer clause structures, though some nonsensical phrases still existed:

> *The applicant submitted that the national authorities had violated their rights under Article 21 of the EC Treaty. It may inform the opinion that has rejected their observations...*

RQ4: Future Improvements

While Case 4 showed clear gains, future improvements may still be possible through:

- Larger model size or deeper architecture
- Gradient checkpointing to allow longer contexts
- Data cleaning to reduce noise in legal phrasing
- **Architecture**: deeper model (more layers), higher embedding dimension, and more attention heads to capture complex legal reasoning.

5 Comparative Analysis Across All Cases

5.1 Performance Comparison

Table 7: Comparative Performance Across All Cases

Metric	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Final Train Loss	4.4413	4.4455	4.2182	3.5799
Final Val Loss	4.6690	4.6373	4.4343	3.7828
Training Time	∼49 min	∼90 min	∼80min	60min
Convergence Speed	Fast	Plateau after 10k	Very stable	Highly Coherent
Text Quality	Partial	Partial+	Improved	gradual Improvement

5.2 Key Findings

- Increasing model size (Case 3 vs Case 1/2) consistently improves text fluency and validation loss.
- Extending training iterations beyond 5000 improves performance (Case 4 vs Case 3), especially with larger models.
- Diminishing returns observed when extending training for small models (Case 2 vs Case 1).
- Larger block sizes help the model learn longer-range legal sentence structures.
- Cyclic learning rate was stable across all cases, supporting extended training.
- Even small language models can capture legal tone and terminology, though deeper semantic coherence requires larger models and longer training.

6 Discussion

The baseline Case 1 experiment reveals several important insights about training domain-specific language models on legal text. The consistent decrease in both training and validation loss indicates effective learning, though the final performance suggests that 5000 iterations may be insufficient for achieving high-quality legal text generation.

The generated text demonstrates partial understanding of legal language patterns, correctly employing formal structures and terminology while struggling with semantic coherence. This suggests that while the model captures surface-level legal language features, deeper semantic understanding requires extended training or architectural modifications.

Limitations Observed:

- Limited training iterations may prevent full convergence
- Block size constraints may limit context understanding
- Model capacity may be insufficient for complex legal reasoning

7 Conclusion

This study explored the process of building a domain-specific small language model for legal text generation using the EURLEX dataset. We evaluated four configurations:

- **Case 1**: 5k iterations baseline — captured legal terms but lacked coherence.
- **Case 2**: 20k iterations — plateaued after 10k iterations with no significant further gain.
- **Case 3**: 5k iterations with larger model — improved text fluency and structure, better domain performance.
- **Case 4**: 20k iterations with larger model — achieved best validation loss (3.78) and most coherent legal text generation.

Across all cases, the model successfully learned the tone, terminology, and structure of legal language. Limitations included occasional incoherence, sentence fragmentation, and lack of deep reasoning — which can be addressed by scaling the model, extending training, and refining data quality.

The Case 4 model demonstrated that increasing training iterations for a larger architecture can lead to significant improvements in both validation loss and text fluency. This underscores the importance of convergence tuning and architectural scale when building domain-specific LLMs.

The project demonstrates that even a relatively small GPT-like model can achieve reasonable results in specialized domains like legal text with careful configuration and training. Future work will explore larger architectures, multi-GPU training, and integration of legal-specific structural cues to further enhance performance.