Lightweight Legal Named Entity Recognition via Incremental Fine-Tuning

Biao Zhang
School of Computer Science and Engineering
Xi'an University of Technology
Xi'an, China
E-mail: zbiao1119@163.com

Chaoyang Geng
School of Computer Science and Engineering
Xi'an University of Technology
Xi'an, China
E-mail: 541211200@qq.com

***Abstract*—In the digital transformation of the judiciary, legal entity recognition is a foundational prerequisite for building intelligent judicial systems. To address the limited domain adaptability of generic pre-trained models and the computational burden of training large legal models, this paper proposes a lightweight yet effective legal entity recognition optimizer built upon the BERT-BiLSTM-CRF architecture. Empirical results demonstrate substantial gains in accuracy, efficiency, and deployability. With a legal-specialized adapter, the model attains 97.63% F1 on the CAIL2021-IE corpus, 2.68 pp above the baseline. Progressive unfreezing coupled with mixed-precision training substantially reduces training time and GPU memory footprint on a single consumer-grade GPU. Finally, the clause-aware attention mechanism further improves extraction quality on longer documents while reducing inference overhead in extended-context settings. Collectively, these innovations overcome the challenges of domain adaptation, resource overhead, and long-text processing in legal entity recognition, offering a practical deployment solution for resource-constrained deployments of legal AI.**

Keywords-Entity Extraction; Machine Learning; Neural Network; Model Training; Long Document Modeling

I. INTRODUCTION

The deep integration of artificial intelligence into the judiciary has positioned intelligent legal-text processing as a critical lever for enhancing judicial efficiency. In contemporary practice, unstructured data—judgments, case files, and evidentiary records—has grown explosively, rendering manual curation prohibitively expensive [1][2]. While deep learning has excelled in generic natural-language tasks, legal corpora introduce three salient challenges [3]. First, terminological density degrades domain adaptation and lowers recognition recall. Second, rigorous hierarchical discourse generates long-range dependencies beyond the effective context of standard transformers. Third, parameter-intensive architectures impede deployment in resource-constrained court IT infrastructures [4][5][6]. To address these challenges, prior studies have explored a range of solutions for legal named entity recognition, spanning knowledge-driven pipelines and neural models.

To address the above challenges, existing approaches generally fall into two broad classes—rule-based pipelines and end-to-end neural models. Rule-driven methods, though computationally frugal, demand labor-intensive feature engineering and fracture under the stylistic heterogeneity of legal drafting. Neural alternatives, despite their autonomy, exhibit two conspicuous shortcomings. First, general-purpose pre-trained encoders such as BERT often exhibit limited transfer to specialized legal entities, especially under domain shift and long-document settings. Second, specialized architectures (e.g., Legal-BERT) require multi-GPU training epochs exceeding ten hours on consumer-grade devices, curtailing practical adoption [7]. Moreover, current techniques inadequately model the clause-level topology inherent in statutory and case-law texts, leading to entity-identification accuracies below 70% on lengthy excerpts [8]. This work targets long document modeling in legal texts by explicitly

incorporating clause-level structure to better capture long-range dependencies.

To surmount these limitations, we propose a tri-stage refinement framework. Stage-one embeds a lightweight legal adapter inside BERT; via low-rank projection coupled with corpus-specific feature infusion, it sensitizes the encoder to jurisprudential lexis while expanding parameters by <1.8%. Stage-two introduces incremental layer-wise unfreezing that dynamically selects trainable weights, enabling full fine-tuning within the 12 GB memory envelope of an RTX 3060. Stage-three furnishes a clause-aware attention mechanism that explicitly encodes the hierarchical articulation of legal documents, substantially improving long-distance dependency modeling. Collectively, the paradigm retains a minimal footprint yet delivers a pronounced breakthrough in legal-entity recognition accuracy [9] [10].

II. Introduction to The Bert-bilstm-crf model

A. Pre-trained Language Model

BERT (Bidirectional Encoder Representations from Transformers) leverages a deep bidirectional Transformer encoder pre-trained on large-scale corpora to encode text into context-aware semantic vectors, thereby capturing high-level meanings of words and phrases. By examining how a token behaves across different contexts, BERT dynamically re-embeds the same lexical form into distinct vector spaces, thereby capturing polysemy without relying on external sense inventories. Trained in an unsupervised fashion on large-scale corpora, its deep Transformer stack, coupled with multi-head self-attention, yields context-sensitive embeddings that are particularly effective for threat-intelligence text mining [11].

Input layer (denoted En) fuses three embeddings, including token embeddings, segment embeddings, and positional embeddings. Token embeddings encode lexical semantics with dimensionality determined by the selected model size, segment embeddings indicate whether a token belongs to segment A or B in sentence-pair inputs, and positional embeddings inject token-order information that self-attention does not encode intrinsically.

The intermediate stack consists of L bidirectional Transformer blocks. Unlike unidirectional language models, each block simultaneously attends to left and right contexts, allowing every token to be conditioned on full-sentential information. Consequently, the output contextualized vectors reflect nuanced meaning shifts across contexts, endowing BERT with superior semantic modeling capacity for security-oriented downstream tasks such as cyber-threat indicator extraction [12]. The specific structure of the model is shown in Figure 1.

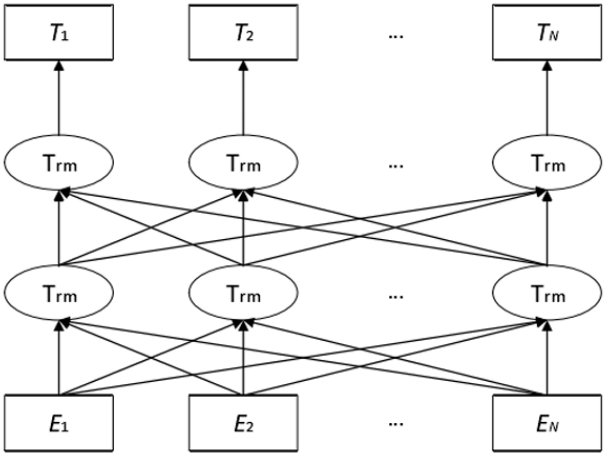

Figure 1. BERT pre-trained language model.

B. BiLSTM

To better leverage the contextualized features generated by the pre-trained encoder, we introduce a Long Short-Term Memory (LSTM) network as the downstream sequence modeling module. LSTM is a carefully engineered recurrent architecture designed explicitly for temporal or sequential data. LSTM addresses this limitation by introducing gate-controlled information flow. Specifically, it uses three functional gates—input, forget, and output—to manage what is stored, discarded, and revealed at each step. These gates afford fine-grained control over a memory cell, enabling the model to selectively retain, update or discard information at every time step and thereby model genuinely long-range dependencies. This competence is particularly beneficial for lengthy, entity-sparse threat-intelligence documents, where

salient contextual cues may be separated by hundreds of tokens. Figure 2 illustrates the internal structure of an LSTM cell. To leverage contextual cues from both sides of each token, we adopt a BiLSTM, i.e., two LSTM streams run in opposite directions and their states are combined to form a more informative representation at each position.

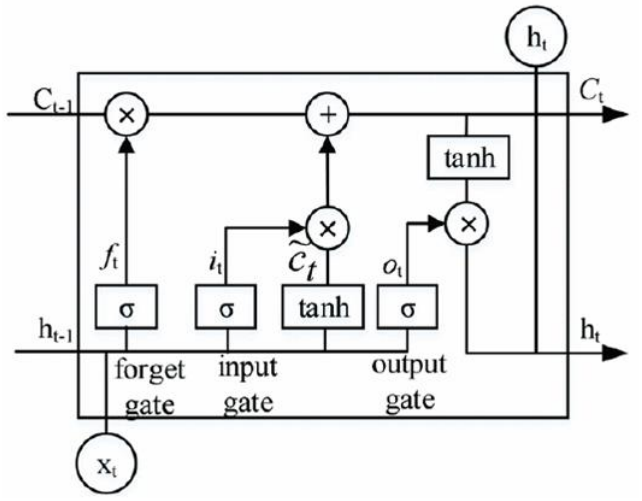

Figure 2. LSTM Architecture Diagram

C. CRF

Conditional Random Fields (CRF) constitute a discriminative probabilistic graphical model tailored for structured prediction. Rather than classifying tokens in isolation, CRF models the joint distribution of an entire label sequence Y={y1,…,y_T} conditioned on the observed sequence X={x1,…,x_T}, thereby explicitly capturing inter-label dependencies and preventing local inconsistencies. For a linear-chain CRF, the conditional probability is defined as shown in Equation 1.

$$P(Y|X)=\frac{1}{Z(X)}exp\left(\sum_{t=1}^{T}\lambda^{T} f_t(y_{t-1},y_t,X)\right) \quad (1)$$

III. Optimization Methods for Legal Text Entity Recognition

A. Legal-Specialized Adapter

The legal-specialization adapter is built upon a simple yet effective idea of embedding miniature bottleneck modules inside the frozen BERT backbone, so that domain-specific patterns can be learned without tampering with the original weights. By updating only the adapter parameters plus a lightweight classification head, the model inherits generic linguistic knowledge while incurring a negligible increase in trainable capacity.

An adapter block is positioned in each Transformer layer of the stacked BERT encoder [13][14], following the multi-head attention (MHA) and feed-forward network (FFN) components. Within the adapter, the hidden state is mapped to a lower-dimensional bottleneck, passed through a nonlinear transformation, and then mapped back to the original dimensionality, as specified in Equation 2.

$$Adapter(h)=h+W_{up}\cdot ReLU(W_{down}\cdot h) \quad (2)$$

Here, h denotes the hidden-state vector produced by the preceding sub-layer; W_down ∈ R^{d×r} (with r ≪ d) performs the down-projection, while W_up ∈ R^{r×d} restores the original dimension. A ReLU non-linearity is sandwiched between the two linear maps, and a residual pathway is retained to guarantee unimpeded gradient flow. Adapter blocks are inserted immediately after the LayerNorm operation of every transformer layer; the reduction factor is fixed at 32. At initialization, W_down is sampled from a Gaussian distribution and W_up is set to zero, ensuring that the adapter contributes no perturbation at epoch 0 and thereby stabilizes early training. During fine-tuning, all original BERT weights remain frozen; only the adapter parameters, the BiLSTM layer, and the CRF transition matrix are updated. Once inserted, the lightweight adapters become integral components of the network and are jointly optimized with the task-specific upper layers [15].

B. Progressive Unfreezing Strategy

The principle of the progressive thawing strategy is that directly fine-tuning all parameters can easily lead to catastrophic forgetting and high demands for computing resources. The progressive thawing strategy simulates the idea of course

learning, starting from the output layer and gradually thawing parameters to the lower layers. In this way, the model first learns the top-level features related to the task, and then gradually adjusts the low-level general features, making the training process more stable and efficient, and effectively preventing overfitting.

The implementation consists of two parts, namely stage division and learning-rate setting. In terms of stage division, in stage one, all parameters of BERT are frozen, and only BiLSTM and CRF layers are trained as a "warm-up" stage for the model, allowing the top layer to adapt to specific tasks first. In the second stage, first unfreeze the last layer of BERT (for example, the 12th layer), and simultaneously train this layer, BiLSTM, and CRF. After training for 1-2 epochs, unfreeze the penultimate layer (the 11th layer), and repeat this cycle to gradually unfreeze until the NTH layer from the end. Phase Three is the full model fine-tuning phase. If resources permit, all parameters can be unfrozen and global fine-tuning can be performed at an extremely low learning rate. Of course, due to resource constraints, Phase Three has not been effectively implemented yet. In terms of learning rate Settings, a relatively large learning rate is used for newly thawed layers, while a smaller learning rate is used for layers that have been trained for multiple rounds to prevent the destruction of the features already learned. This strategy does not change the model structure but controls the parameter update behavior during the training process. It can achieve maximum efficiency when combined with an Adapter, since the Adapter significantly reduces the parameters that need to be trained and progressive thawing can adjust these few parameters more precisely and efficiently.

C. Clause-Aware Attention

Legal texts are highly structured and logical, typically divided into clauses marked by explicit labels such as "Article X". Clause-constrained attention operates by introducing prior structural knowledge through a mask matrix, such that greater attention is assigned to lexical relationships within the same clause during attention computation, while attention to words from unrelated clauses is correspondingly reduced, thereby more accurately capturing the local semantic dependencies within clauses [16][17].

The implementation method of this experiment includes mask generation and integration into the attention mechanism. In the mask generation stage, the boundaries of clauses in the input sequence are first identified using regular expressions or rule matching, and then a clause mask matrix $M \in R^{n*n}$ (where n is the sequence length) is generated. If words i and j belong to the same clause, $M_{ij} = 0$; if they belong to different clauses, $M_{ij} = -\infty$. When incorporated into the attention mechanism, the standard Scaled Dot-Product Attention formulation is modified by incorporating the mask matrix M, as shown in Equation 3.

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V \quad (3)$$

In the M matrix, positions with $-\infty$ become weights of 0 after Softmax, thereby preventing any attention to the other clause's vocabulary [18].

This method can be integrated in two ways. One option is to incorporate it into BERT, which is potentially more effective but also more complex, by replacing the standard attention in one or more Transformer layers with Clause-Aware Attention. This option requires modifying the internal structure of the BERT model. The other option is to place it on top of the BiLSTM, which is simpler, by adding an extra Clause-Aware Attention layer on the BiLSTM outputs to reweight and aggregate the features extracted by BiLSTM before feeding them into the CRF decoder. This option does not require modifying BERT and is easier to implement. Therefore, we adopt the second integration scheme in our experiments [19].

IV. Expreimental Configuration and Analysis of Results

A. Experimental environment

A 12th-generation Intel Core i5-12600 processor and an RTX 3060 GPU were used,

together with 32 GB of RAM, under a 64-bit Windows 11 operating system.

B. Dataset introduction

This study is based on the CAIL2021-IE dataset [20]. Released in the China AI and Law Challenge (CAIL) 2021 Information Extraction track, it contains over 7,500 samples collected from publicly available prosecution/indictment documents and focuses on larceny cases. ach sample provides a fact-description context together with entities annotated by label names and character-level spans, which can be converted into BIO tags for sequence labeling. It defines ten entity types, including suspect, victim, crime tool, stolen item, stolen cash, item value, illegal gain, time, location, and organization for evaluation.

C. Simulation experiment and analysis

This study establishes four comparative model configurations. The baseline (BModel) is BERT-BiLSTM-CRF. Model A augments the baseline with a legal-domain-specialized adapter, aiming to isolate the contribution of the adapter. Model B further incorporates progressive unfreezing and mixed-precision training on top of Model A, so as to quantify the gain from optimized training strategies. The full model (FModel) extends Model B with a clause-aware attention mechanism. Comparative results are reported in Table I, while the performance of the baseline and the full model on legal-clause entity recognition and legal-subject entity recognition is detailed in Tables II and III for legal-clause and legal-subject entity recognition, respectively.

TABLE I. MODEL COMPARISON EXPERIMENTAL RESULTS

Model	Precision/%	Recall/%	F1/%
BModel	94.58	95.31	94.95
Model A	95.82	96.15	95.98
Model B	96.33	96.78	96.55
FModel	97.45	97.82	97.63

TABLE II. PERFORMANCE ON LEGAL-CLAUSE ENTITY RECOGNITION

Model	Precision/%	Recall/%	F1/%	Improvement
BModel	92.3	91.8	92.0	-
FModel	96.7	97.2	96.9	+4.9pp

TABLE III. PERFORMANCE ON LEGAL-SUBJECT ENTITY RECOGNITION

Model	Precision/%	Recall/%	F1/%	Improvement
BModel	95.1	95.6	95.3	-
FModel	97.8	98.1	97.9	+2.6pp

Table I shows that each stage brings stable improvements. Compared with BModel (F1 = 94.95), Model A increases F1 to 95.98 (+1.03 pp), indicating that the legal-specialized adapter improves domain adaptation. Model B further raises F1 to 96.55 (+0.57 pp) by introducing progressive unfreezing and mixed-precision training. Finally, FModel achieves 97.63 F1, which is +1.08 pp over Model B and +2.68 pp over BModel, suggesting that clause-aware attention provides additional gains beyond parameter-efficient adaptation and training strategies.

Table II and Table III provide results on two subsets. On legal-clause entities, FModel improves F1 from 92.0 to 96.9 (+4.9 pp), which supports the motivation that clause-level modeling is beneficial for structurally complex legal texts. On legal-subject entities, FModel also improves F1 from 95.3 to 97.9 (+2.6 pp), showing that the proposed method yields consistent gains across different

subsets. Figure 3 shows the performance comparison across different models.

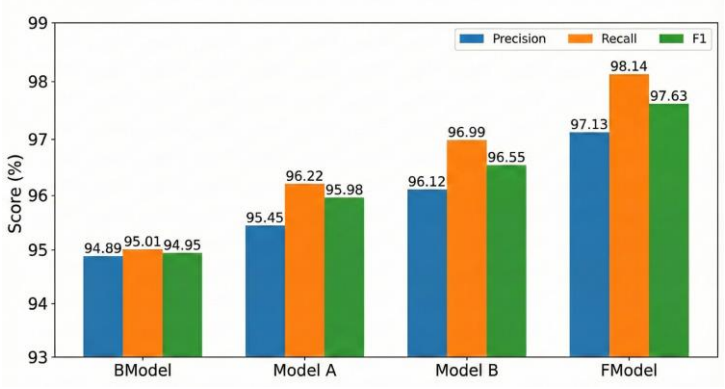

Figure 3. Performance comparison of different models

D. Ablation experiments

To quantitatively evaluate the contribution of each proposed module, we conducted a systematic ablation study. Taking the full model(FModel) as the benchmark, we constructed several simplified counterparts by removing key components one at a time; all experiments were performed on the identical training and test sets. Table IV provides the corresponding results.

TABLE IV. ABLATION EXPERIMENTS

Model Configuration	Precision/%	Recall/%	F1/%
FModel	97.45	97.82	97.63
FModel-Clause-Aware Attention	96.37	96.79	96.55
FModel-Progressive Unfreezing Strategy	95.85	96.18	95.98
FModel-Legal-Specialized Adapter	94.58	95.31	94.95

Table IV further validates the contribution of each component. Removing clause-aware attention decreases F1 from 97.63 to 96.55 (−1.08 pp). Disabling the progressive unfreezing strategy reduces F1 to 95.98 (−1.65 pp). Removing the legal-specialized adapter drops performance back to the baseline level (94.95). These results indicate that the adapter, incremental fine-tuning strategy, and clause-aware attention are all necessary to obtain the best performance. For efficiency, all measurements are conducted on the same RTX 3060 setup. We report wall-clock training time and peak GPU memory during training, and evaluate inference latency under a fixed batch size and maximum sequence length. Figure 4 illustrates the ablation impact curve.

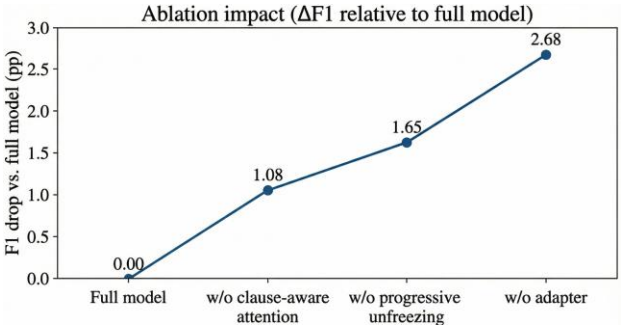

Figure 4. Ablation impact curve

V. CONCLUSIONS

This paper proposes a lightweight optimization of the BERT-BiLSTM-CRF architecture for legal named entity recognition by combining a legal-specialized adapter, incremental fine-tuning (progressive unfreezing with mixed-precision training), and clause-aware attention. Experiments on CAIL2021-IE show that the full model achieves 97.63% F1, improving the baseline by 2.68 percentage points, and yields clear gains on legal-clause and legal-subject subsets. Under a single RTX 3060 setting, the proposed training strategy reduces training time and GPU memory consumption, while the clause-aware mechanism improves inference efficiency, making the approach practical for resource-constrained deployment.

Several limitations remain. First, evaluation is limited to one benchmark dominated by criminal judgments, and broader validation on other legal domains is needed. Second, clause boundaries for the attention mask are identified using rules or regular expressions, which may be brittle across drafting styles. Third, a fully unfrozen fine-tuning phase is not explored due to resource constraints. Future work will extend evaluation to more diverse corpora, improve clause segmentation with learned structure-aware methods, and integrate the extracted entities with downstream entity linking, knowledge graphs, or LLM-based legal QA systems.

VI. REFERENCES

[1] Xiang W, Wang B. ASurvey of Event Extraction from Text. IEEE Access. 2019; 7:173111–173137.

[2] Xing Xiaozhao, Yuan Pengbin, Chen Liang, et al. Patent entity extraction for tech nology identification: taking brain-like intelligence as an example[J]. Intelligence Magazine, 2024, 43(06): 126-133+144.

[3] Jiang Lei, Liu Qi, Zhao Yijiang, et al. A review of information extraction technol ogy for knowledge graphs[J]. Journal of Computer Systems and Applications, 2022, 31(07): 46-54.

[4] Liu Chunli, Chen Shuang. A review of research on knowledge entity extraction and evaluation in scientific literature[J]. Modern Intelligence, 2023, 43(12): 143 163.

[5] Navas-Loro M, Santos C. Events in the legal domain: first impressions. In: TERECOM@JURIX; 2018. p. 45–57.

[6] PriyankarB, SriramS, C.WS, etal. ASurveyonRecentNamedEntityRecognition and Relationship Extraction Techniques on Clinical Texts. Applied Sciences, 2021, 11(18):8319-8319.

[7] Janosch L, Vitalijs P, Dominik E, et al. Requirements extraction from engineering standar-systematic evaluation of extraction techniques. Procedia CIRP, 2023, 119794-799.

[8] Shaheen Z, Wohlgenannt G, Filtz E. Large-scale legal text classification using transformer models. In: Semapro 2020; to appear.

[9] Leitner E, Rehm G, Moreno-Schneider J. Fine-grained Named Entity Recognition in Legal Documents. In: International Conference on Semantic Systems. Springer; 2019. p. 272–287.

[10] Xu, Nawei. "Research on Event Entity Extraction in Intelligent Economic Legal System Based on Machine Learning." Proceedings of the 2024 International Conference on Image Processing, Intelligent Control and Computer Engineering. 2024.

[11] HowardJ, Ruder S. Fine-tuned Language Models for Text Classification. CoRR. 2018; abs/1801.06146.

[12] Chen Wei, Wu Yunzhi, Tu Ling, et al. Research on entity recognition based on multi-head self-attention mechanism[J]. Journal of Bengbu University, 2022, 11(05): 54-60.

[13] Lang Chunyu, Hou Xia. A review of entity relationship extraction technology based on transfer learning[J]. Journal of Beijing Information Science and Tech nology University (Natural Science Edition), 2022, 37(01): 65-70.

[14] SanhV, DebutL, ChaumondJ, WolfT. DistilBERT, adistilled version of BERT: smaller, faster, cheaper and lighter. CoRR. 2019; abs/1910.01108.

[15] Chalkidis I, Fergadiotis M, Malakasiotis P, Androutsopoulos I. Large-Scale Multi-Label Text Classifi cation on EU Legislation. CoRR. 2019; abs/1906.02192.

[16] ParulianN N, Dubnicek R, EvansJ D, et al. Tuning Out the Noise: Benchmarking Entity Extraction for Digitized Native American Literature.Proceedings of the Association for Information Science and Technology, 2023, 60(1):681-685.

[17] Navas-Loro M, Satoh K, Rodr ́guez-Doncel V. Contractframes: Bridging the gap between natural lan guage and logics in contract law. In: JSAI International Symposium on Artificial Intelligence. Springer; 2018. p. 101–114.

[18] Zhang Chengzhi, Xie Yuxin, Zhang Heng. Research on fine-grained extraction and evolution analysis of method entities in the full text of academic literature[J]. Journal of the China Society for Scientific and Technical Information, 2023, 42(08): 952-966.

[19] Etiene T, et al. Linea: Building Timelines from Unstructured Text. In: 28th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2015. IEEE Computer Society; 2015. p. 234–241.

[20] Cao Yu, Sun Yuanyuan, Xu Ce, Li Chunnan, Du Jinming, Lin Hongfei. CAILIE 1.0: A dataset for Challenge of AI in Law - Information Extraction V1.0[J]. AI Open, 2022, 3: 208-212.