Hierarchical Long-Document Summarization using LED for Legal Judgments

Reshma Sheik[1] **Noah John Puthayathu**[1] **Fathima Firose A**[2] **Jonathan Paul**[1]
[1]TKM College of Engineering, Kollam [2]Thiagarajar College of Engineering, Madurai
rezmasheik@gmail.com,230979@tkmce.ac.in,
fathimafiroseofficial@gmail.com,230685@tkmce.ac.in

Abstract

This paper describes our system for the L-SUMM shared task on legal document summarization. Our approach is built on the Longformer Encoder-Decoder (LED) model, which we augment with a multi-level summarization strategy tailored for legal documents that are substantially longer than typical transformer input limits. The system achieved competitive performance on the legal judgment summarization task through optimized training strategies, including gradient accumulation, Adafactor optimization, and hyperparameter tuning. Our findings indicate that combining hierarchical processing with strategically assigned global attention enables more reliable summarization of lengthy legal texts.

1 Introduction

Summarizing legal documents presents unique challenges due to the length, complexity, and specialised terminology of legal texts. Judicial rulings can span many thousands of words and include intricate argumentation, citations, and procedural information that should be retained in any condensed representation. Recent breakthroughs in transformer-based models have made significant progress in handling long documents more effectively (Lewis et al., 2019; Zhang et al., 2020). The Longformer Encoder-Decoder (LED) architecture extends transformers to handle sequences up to 16,384 tokens through efficient attention patterns (Beltagy et al., 2020). However, many legal judgments exceed even this extended context window.

We present a hierarchical summarization system that combines LED's long-context capabilities with a chunk-and-aggregate approach for extremely long documents. We evaluate our system on the L-SUMM shared task using the InLSum dataset. Our contributions include:

- A robust preprocessing pipeline handling diverse legal text formatting
- Hierarchical summarization for documents exceeding model capacity
- Extensive hyperparameter optimization for legal domain adaptation

2 Related Work

2.1 Efficient Attention Mechanisms for Long Sequences

Traditional transformer models are limited to 512 or 1024 tokens because their self-attention mechanism scales quadratically with sequence length (Vaswani et al., 2017). Several approaches have been proposed to extend this capacity, including sparse attention patterns (Zaheer et al., 2020), retrieval-augmented methods (Lewis et al., 2020), and hierarchical architectures (Cohan et al., 2018).

The Longformer model employs a mixed attention pattern, combining local window-based attention with global tokens to efficiently model long sequences. These architectures have proven to be effective for long-document understanding and summarization tasks.

2.2 Legal Document Processing

Legal NLP has seen increasing interest, with shared tasks focusing on case outcome prediction, statute retrieval, and summarization (Kano et al., 2018; Chalkidis et al., 2020). Legal texts require domain-specific handling due to their length, formal language, and citation structure.

Previous surveys highlight the challenges of summarizing lengthy legal judgments, emphasizing the need for specialized models capable of handling complex reasoning and domain-specific terminology (Kanapala et al., 2019).

2.3 Indian Legal Document Summarization

Automatic summarization and structural analysis of Indian legal judgments have attracted attention

Proceedings of the 1st Workshop on NLP for Empowering Justice (JUST-NLP 2025), pages 191–195
December 24, 2025 ©2025 Association for Computational Linguistics

in recent years, as the large volume and complexity of judgments pose challenges to accessibility and comprehension. Previous work such as Bhattacharya et al. (2019) studied rhetorical role classification of sentences in judgments from the Indian Supreme Court, allowing the segmentation of legal documents into semantically meaningful segments such as facts, issues, reasoning, and rulings, which is a useful preprocessing step for summarization and retrieval. Bhattacharya et al. (2021) proposed an unsupervised summarization method (DELSumm) for Indian case documents, showing promising ROUGE scores without the need for large annotated datasets. Building on dataset creation efforts, Parikh et al. (2021) released a corpus of over 10,000 Indian judgments paired with handwritten summaries and developed weakly supervised summarization baselines. Our work extends these efforts by combining hierarchical processing with long-context transformer architectures to handle extremely long judgments, a challenge common in the Indian legal domain.

3 System Description

Our system employs a hybrid approach that adapts to document length. Figure 1 provides an overview of the complete pipeline, showing both the direct summarization path for shorter documents and the hierarchical approach for longer ones.

3.1 Model Architecture

We employ the LED-large-16384 model[1] as our base architecture. LED extends the Longformer's efficient attention mechanism to seq2seq tasks, combining:

- **Local attention**: Sliding window attention with window size 512
- **Global attention**: Selected tokens that attend to all positions
- **Encoder-decoder structure**: Enabling abstractive generation

Our implementation configures global attention on the first 64 tokens and then periodically every 384 tokens throughout the document. This pattern was selected after examining the structure of the legal judgments in the dataset, where important elements such as party names, issue statements, statutory references, and section headings recur at regular intervals. Assigning global attention at these periodic positions ensures that structurally significant segments are consistently captured.

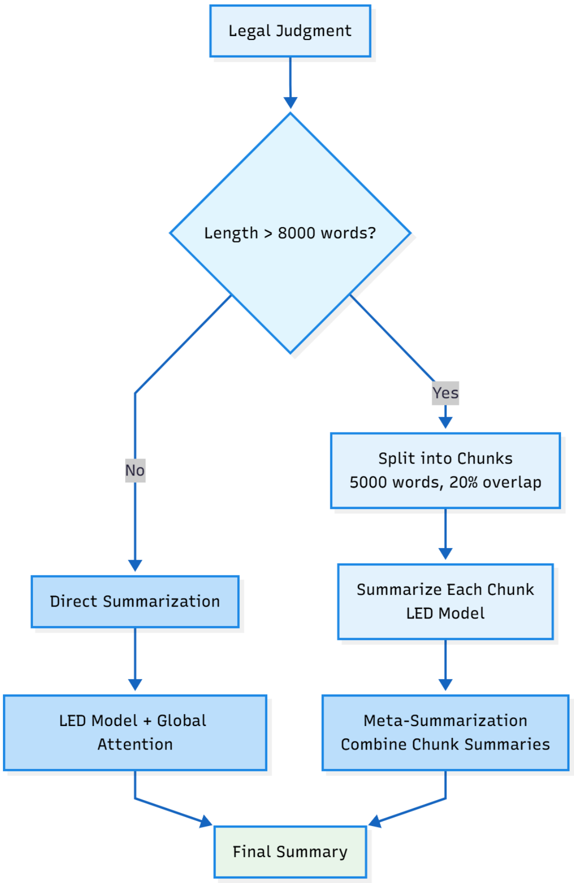

Figure 1: System pipeline showing adaptive processing based on document length.

3.2 Data Preprocessing

Legal judgments contain various formatting artifacts that require careful preprocessing:

1. **Page number removal**: Patterns like [Page No. X] are stripped
2. **Whitespace normalization**: Excessive newlines and spaces are compressed
3. **Column name standardization**: All metadata fields are lowercased for robust schema handling

The preprocessing ensures clean input while preserving the semantic content and structure of legal arguments.

[1] https://huggingface.co/allenai/led-large-16384

3.3 Hierarchical Summarization

For documents exceeding 8,000 words, we employ a two-stage hierarchical approach:

Stage 1: Chunk Summarization The document is divided into overlapping chunks of 5,000 words with 20% overlap (1,000 words). Each chunk is summarized independently with the prompt: "Summarize this legal judgment." This produces intermediate summaries capturing key information from each document section.

Stage 2: Meta-Summarization The chunk summaries are concatenated and processed with a meta-prompt: "Combine these summaries into one coherent summary." This stage synthesizes a final unified summary that maintains coherence throughout the document.

For documents under 8,000 words, direct single-pass summarization is used without chunking.

Design Justification The chunk size of 5,000 words was determined through preliminary experimentation to balance context preservation and computational feasibility. A 20% overlap mitigates boundary effects, ensuring continuity across legal sections that often span multiple paragraphs.

3.4 Training Configuration

We fine-tune LED-large on the training set with the configuration shown in Table 1.

Parameter	Value
Optimizer	Adafactor, learning rate 3e-5
Batch size	1 per device, 16 gradient accumulation steps (effective 16)
Training epochs	15
Input length	9,192 tokens maximum
Output length	400–768 tokens
Warmup	15% of training steps
Weight decay	0.01
Mixed precision	FP16 on GPU
Checkpointing	Every 500 steps, retain last 3

Table 1: Training configuration for fine-tuning LED-large on the InLSum dataset.

Gradient checkpointing is enabled to reduce memory consumption, allowing larger effective batch sizes through gradient accumulation. The final model, after 15 epochs of training, is used for inference on the test set.

3.5 Generation Strategy

The beam search parameters used during inference are summarized in Table 2.

Parameter	Value
Beam size	8
Length penalty	1.2 (encourages longer summaries)
Repetition penalty	1.5
No-repeat n-gram size	4
Early stopping	Enabled

Table 2: Beam search parameters used during inference.

Post-processing removes only strict sentence-level duplicates by normalizing and hashing full sentences. This ensures that legally meaningful repetitions are preserved while eliminating redundant generated text.

4 Experimental Setup

4.1 Dataset

The InLSum dataset[2] consists of Indian legal judgments paired with reference summaries written by experts. Table 3 provides detailed token-level statistics for training, validation, and test splits using the LED tokenizer.

4.2 Evaluation Metrics

Model quality was assessed using ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002). ROUGE-2 and ROUGE-L evaluate overlap and sequence consistency, while BLEU measures n-gram precision and fluency characteristics.

The final ranking is determined by the average of ROUGE-2, ROUGE-L, and BLEU scores. This combined metric balances different aspects of summary quality: bigram overlap (ROUGE-2) captures content selection, sequence matching (ROUGE-L) evaluates structural coherence, and BLEU evaluates overall n-gram precision and fluency.

4.3 Implementation Details

This system is implemented in PyTorch using the HuggingFace Transformers library. Training was

[2] https://exploration-lab.github.io/JUST-NLP/task/

Statistic	Value
Training samples	1200 pairs
Validation samples	200 judgments
Test samples	400 judgments
Judgment length range	229 – 329,814 tokens
Median judgment length	4,360 tokens
Average judgment length	10,926 tokens
Summary length range	39 – 2,689 tokens
Median summary length	672 tokens
Average summary length	708 tokens
Domain	Indian legal court judgments
Tokenizer used	`allenai/led-large-16384`

Table 3: Token-level statistics of the InLSum dataset using the LED tokenizer.

conducted on Nvidia RTX A6000. Key implementation choices include:

- Schema-robust data loading handling column name variations
- Safe iteration over DataFrame rows during inference
- Runtime fallback for out-of-memory cases using truncation
- Comprehensive error handling for edge cases

5 Results and Analysis

5.1 Main Results

Table 4 presents the system's performance on the InLSum test set. The hierarchical LED-based approach achieves competitive results across all metrics, with ROUGE-2 of 29.62, ROUGE-L of 28.56, and BLEU of 21.67, yielding an average score of 26.62. The obtained metrics indicate that the proposed system extracts salient legal content and produces fluent summaries.

Our system achieved second rank on the official L-SUMM shared task leaderboard, placing it among the competitive submissions in the shared task. This demonstrates that hierarchical processing, combined with LED's long-context attention, is an effective strategy for summarizing extremely long legal judgments.

Metric	Score
ROUGE-2	29.62
ROUGE-L	28.56
BLEU	21.67
Average	**26.62**

Table 4: Performance of our system on the InLSum test set. All scores are reported as percentages. The average is computed across ROUGE-2, ROUGE-L, and BLEU.

The ROUGE-2 and ROUGE-L scores indicate strong bigram and sequence-level overlap with reference summaries, suggesting our system captures important factual content and legal reasoning patterns. The BLEU score of 21.67 reflects reasonable n-gram precision. The overall average of 26.62 indicates a balanced performance across various evaluation dimensions.

5.2 Error Analysis

Common failure modes include:

- **Citation handling**: Complex citation chains sometimes lose context
- **Multi-party cases**: Cases with numerous parties occasionally conflate identities
- **Procedural details**: The balance between procedural and substantive content varies

6 Conclusion

We presented a hierarchical LED-based system for legal document summarization that effectively handles extremely long judgments through a combination of chunk-level processing and meta-summarization. Using strategic global attention patterns and carefully tuned hyperparameters, our system achieves strong performance on the L-SUMM shared task. The approach successfully captures the structural and semantic complexity of long legal texts.

In future work, we plan to explore the integration of legal knowledge bases for improved citation handling, incorporate multi-task learning with related legal NLP objectives, and investigate adaptive chunking strategies that align with the discourse structure of judgments. Another promising direction is the design of verdict-aware prompting mechanisms to improve the specificity and interpretability of generated summaries. Overall, our results demonstrate that combining long-context architectures with hierarchical summarization pipelines is a practical and effective solution for legal document summarization.

Limitations

This work has several limitations. The hierarchical design, while effective for managing extremely long documents, introduces additional computational overhead and increases inference latency compared to single-pass models. Despite the use of overlapping chunks, segmentation may still split important contextual information, occasionally affecting the continuity of legal reasoning in the final summary. Our system was trained and evaluated exclusively on Indian legal judgments, raising questions about generalizability to other jurisdictions or legal writing styles. Memory constraints restricted batch sizes during training, limiting the extent of hyperparameter exploration and ablation studies. Finally, the meta-summarization stage can sometimes compress information too aggressively, resulting in minor coherence issues in cases involving dense reasoning or complex procedural histories.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Wyner. 2019. Identification of rhetorical roles of sentences in indian legal judgments. In *Legal knowledge and information systems*, pages 3–12. IOS Press.

Paheli Bhattacharya, Soham Poddar, Koustav Rudra, Kripabandhu Ghosh, and Saptarshi Ghosh. 2021. Incorporating domain knowledge for extractive summarization of legal case documents. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 22–31.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. 2019. Text summarization from legal documents: a survey. *Artificial Intelligence Review*, 51(3):371–402.

Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2018. Coliee-2018: Evaluation of the competition on legal information extraction and entailment. In *JSAI International Symposium on Artificial Intelligence*, pages 177–192. Springer.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Vedant Parikh, Vidit Mathur, Parth Mehta, Namita Mittal, and Prasenjit Majumder. 2021. Lawsum: A weakly supervised approach for indian legal document summarization. *arXiv preprint arXiv:2110.01188*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and 1 others. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pages 11328–11339. PMLR.