1 Acknowledgements

I would like to express my gratitude to Professor Hovy for his continued mentorship along this journey and for making this experience enriching and enjoyable. Thank you to all the colleagues who participated in the surveys; your contribution is highly valued and appreciated. Finally, I would like to thank my family for their unwavering support. You are my constant source of inspiration and I could not have done any of this without you.

From Statutes to Summaries: Metrics, Length Constraints, and Model Adaptation in Transformer-Based Legal Text Summarization

Redi Alico

July 24, 2026

Contents

2 Abstract

Automated text summarization systems traditionally rely on architectural heuristics optimized for journalistic media, imposing strict input limits (e.g., 1,024 tokens) that incur a severe ”truncation penalty” when applied to legislative bills, where critical statutory clauses are distributed throughout multi-page documents. This thesis addresses the architectural and evaluation challenges of high-fidelity legal document summarization. By expanding input context windows up to 16,384 tokens using sparse-attention architectures (LED, LongT5) and evaluating output caps at 128 and 256 tokens, this work systematically analyzes the interplay between context boundaries, domain adaptation, and evaluation metric reliability. Evaluation is conducted across a three-tiered framework comprising lexical overlap (ROUGE, BERTScore), reference-free semantic yield (BLANC), and qualitative alignment (human legal experts vs. AI evaluators).

Empirical results demonstrate that coupling a 16,384-token sparse input context with a 256-token generation cap provides the optimal baseline for preserving complex statutory conditions while maintaining human readability. Among the evaluated models, domain-adapted LED-base emerges as the most consistent performer across all three evaluation tiers, outperforming both short-context baselines (BART) and general long-context models (LongT5). Crucially, qualitative evaluation reveals a distinct abstraction penalty: while LED-large earned top ratings from human legal experts and automated metrics, AI-as-a-Judge frameworks penalized its fluid abstractive rephrasings as factual deviations. This misalignment proves that while AI evaluators offer value as rapid screening assistants, they remain unreliable as standalone legal judges. Finally, the practical operational viability of extended-context summarization is demonstrated through LexiBrief, an interactive web tool built to assist policy analysts and compliance professionals in real-world workflows.

3 Introduction

3.1 The Legal Document Anomaly

In the field of Natural Language Processing (NLP), automatic text summarization has achieved remarkable success, driven largely by benchmarks dominated by journalistic media, such as the CNN/DailyMail dataset. However, the architectural heuristics that succeed on news text collapse entirely when applied to legislative and legal documents. While a news article features an inherent 'lead bias' — where journalists front-load the most critical facts into the opening paragraphs — a legislative bill distributes its informational weight evenly across its entire length. A single, critical statutory amendment, hidden definitions clause, or budgetary allocation can appear on the final page of an otherwise mundane multi-page document. For legal professionals, policy analysts, and citizens, keeping pace with this volume of dense text is a logistical impossibility, creating a clear and urgent demand for automated, high-fidelity summarization systems.

The practical consequences of these technological limitations are not merely academic; they introduce severe operational liabilities for organizations tasked with regulatory compliance. According to recent industry data, the sheer velocity and volume of legislative activity have compounded systemic oversight risks, with 58% of legal and policy professionals identifying the "fear of missing a critical statutory update" as their primary day-to-day operational anxiety [1]. When vital amendments, definitions, or compliance exceptions are buried deep within multi-page omnibus bills, manual human review inevitably falls victim to cognitive fatigue. The inability of standard NLP tools to parse these deep-text structures means organizations regularly risk missing buried liabilities, leading to severe regulatory non-compliance, unexpected financial penalties, and compromised strategic planning.

3.2 The Technological Bottleneck & Truncation Penalty

Standard transformer architectures rely fundamentally on the self-attention mechanism, which exhibits a quadratic computational and memory complexity of $O(N^2)$ relative to the input sequence length N. While this scaling factor is entirely manageable for short-form text processing — such as analyzing news articles or consumer reviews — it presents an insurmountable hardware barrier when applied to long-form legal documents. Attempting to pass an entire, unedited legislative bill through a traditional full-attention network causes an exponential explosion in memory usage, quickly exhausting available GPU resources and rendering end-to-end sequence processing computationally impossible.

To maintain model stability and prevent these memory failures, standard natural language processing pipelines deploy a crude architectural workaround: aggressive text truncation. Baseline models like standard BART feature a rigid maximum input threshold of 1,024 tokens. When faced with a document exceeding this limit, the preprocessing script simply discards all trailing text.

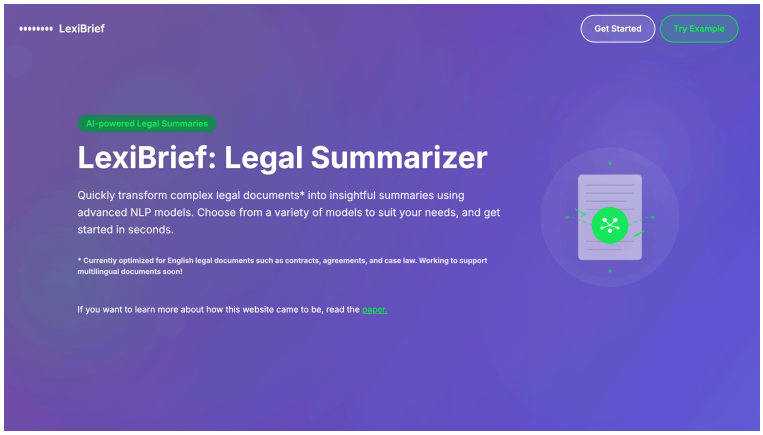
While this approach protects the hardware, it imposes a severe "truncation penalty" when processing legal text. Because legislative documents distribute their informational weight evenly throughout the text rather than front-loading it, blindly slicing a bill at the 1,024-token

mark leaves the underlying model completely blind to the remaining segments. The network is forced to generate a summary based on a highly fragmented input, ensuring that any vital legal provisions or amendments located in the latter half of the document are entirely omitted from the final output.

3.3 The Proposed Solution

To overcome the architectural and evaluation limitations outlined in the preceding sections, this thesis develops an experimental framework that systematically eliminates context-window restrictions while incorporating a semantic-heavy evaluation pipeline. Rather than accepting the traditional truncation penalty, this study leverages recent advancements in sparse-attention and long-context architectures — specifically the Longformer Encoder-Decoder (LED) and LongT5 models. By scaling the input processing capacity up to 16,384 tokens, these networks read lengthy legislative text unbroken, completely preserving the long-range statutory context that standard baselines discard. Furthermore, this work evaluates the distinct compounding benefits achieved when these extended context windows are paired with domain-adapted checkpoints pre-trained explicitly on legal vocabularies.

Recognizing that surface-level text overlap metrics like ROUGE fail to capture true legal comprehension, this study introduces a multi-dimensional evaluation framework. We augment lexical matching with the reference-free BLANC metric, executing Cloze-task document reconstructions to mathematically quantify the functional information yield of our generated text. Finally, to bridge the gap between model training and real-world utility, this work deploys these optimized networks into a functional, interactive web interface, proving the industrial viability of automated legal summarization tools. You can view a screenshot of the website below. The link to the website is provided in the contributions section.



4 Contributions

- Adaptation of transformer-based models for **effective** summarization of long legal texts
- Introduction of a **comprehensive** multi-metric evaluation framework
- **Empirical** analysis of length constraints in summarization
- **Interactive** website incorporating this tool (https://www.lexibrief.live)

5 Related Work

In this section, we will take a look at previous work in the field (papers on the models, evaluation metrics, and techniques) to understand the extent of the work done so far and what this paper adds.

5.1 The evolution of Neural Text Summarization

Text summarization has long been a focal point of computational linguistics, evolving from rigid extractive methods to the more fluid abstractive paradigm. Extractive summarization, analogous to highlighting key sentences in a physical document, is a conservative approach that selects fragments from the source text without modification.[2] While this ensures grammatical correctness, it inherently lacks the synthesis and flow required to condense complex ideas. Consequently, the field has shifted toward abstractive summarization [3], which aims to mimic human cognition by generating novel sentences that paraphrase and consolidate information.

The integration of large-scale pre-training into this field was significantly advanced by the work of Liu & Lapata [4], who sought to address the limitations of the Bidirectional Encoder Representations from Transformers (BERT). While the original BERT model excelled at natural language understanding (NLU) through its encoder-only architecture, it lacked the ability to generate sentences necessary for abstractive tasks.

To bridge this gap, Liu & Lapata introduced BERTSUM, a framework that adapts BERT for document-level representations. Their primary innovation lay in modifying BERT's input architecture: while the original model utilizes a single [CLS] token for the entire sequence, BERTSUM inserts a [CLS] token at the start of every sentence. This, combined with interval segment embeddings, allows the model to learn distinct, sentence-level representations rather than a single global embedding. A closer look at the figure below shows the structural difference between the original BERT and BERTSUM.

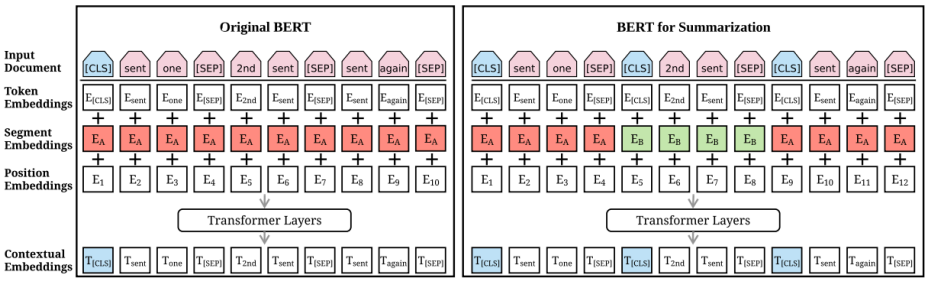
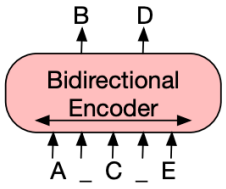


Figure 1: Architecture of the original BERT model (left) and BERTSUM (right). The sequence on top is the input document, followed by the summation of three kinds of embeddings for each token. The summed vectors are used as input embeddings to several bidirectional Transformer layers, generating contextual vectors for each token. BERTSUM extends BERT by inserting multiple [CLS] symbols to learn sentence representations and using interval segmentation embeddings (illustrated in red and green color) to distinguish multiple sentences.

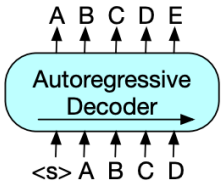
For the extractive task, BERTSUM leverages these sentence-specific embeddings to rank and select source sentences. However, the more significant architectural leap occurs in BERTSUM-Abs (the abstractive variant). Recognizing that an encoder alone cannot generate new text, the authors integrated a Sequence-to-Sequence (Seq2Seq) decoder, adopting the structural logic popularized by See et al. (2017). [5]

The fundamental challenge in this approach was the architectural asymmetry: the model paired a sophisticated, pre-trained encoder with a randomly initialized decoder. To prevent the decoder from conflicting with the encoder's pre-existing knowledge during fine-tuning, Liu & Lapata implemented a dual-optimizer strategy. By applying a smaller learning rate to the encoder and a higher one to the decoder, the framework allowed the decoder to rapidly learn the mechanics of language generation while the encoder refined its representation of the source document.

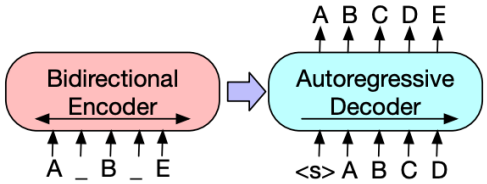
The next big leap would come from the 2019 paper by Mike Lewis and his co-authors. They shifted from a pre-trained encoder and a 'glued on' decoder (BERTSUM) to a fully pre-trained encoder-decoder architecture, which is called Bidirectional & Autoregressive Transformer (BART).[6] BART merges BERT and GPT by acknowledging the limitations of each architecture and binding them to complement each other. BERT on its own can understand natural language very well. Similarly, GPT on its own can generate text, but it does so only by learning context from one direction. Pairing it with BERT resulted in a masterful display of innovation in the field. We now have an architecture which not only understands text remarkably well, but also understands representations bidirectionally. This results in a tool (though incomplete) for text summarization, specifically tailored to abstractive methods.



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



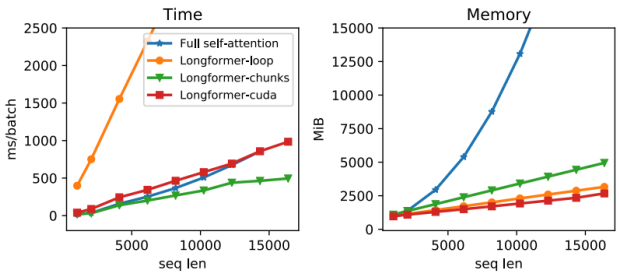
(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Despite the architectural success BART had, it (and its predecessors) faced a fundamental limitation: token length cap. Because these architectures rely on what is called the Self-Attention mechanism, they are capped at 512 or 1024 tokens.[7] Since the tokens in a sequence must attend to every other token, Transformers possess computational complexity of $O(n^2)$, where n represents sequence length. This in turn means that whenever we want to double our input sequence length, we'll have to quadruple memory usage. This limitation prevents models like BART to be used for summarization of longer texts such as legal transcripts, books, or even technical reports, unless aggressive truncation is applied.

5.2 Scaling to Long-Form Documents

To extend the limit of tokens that can be taken into context, computational linguists were forced to once again make changes to current architectures. Just like moving from BERT to BERTSUM, the breakthrough from BART would be the modification of current bottlenecks, in this case self attention. To that end, Iz Beltagy, Matthew E. Peters, and Arman Cohan of the Allen Institute for Artificial Intelligence (Ai2) developed the Longformer. While captivating in its structure to say the least, a variant of the Longformer is what we're actually after. The Longformer-Encoder-Decoder (LED) [8] is the extension for long-document generative tasks, performing notably well on the arXiv summarization dataset.

The Longformer uses an attention mechanism which scales linearly instead of quadratically with respect to sequence length, thus allowing for a much larger token cap. It does this by introducing a 'sliding window attention pattern', which I like to link to convolutional neural networks (CNNs). [9] CNNs are architectures used extensively for tasks involving images, and contain so-called convolutional layers (among others). These layers effectively employ a filter (kernel), which slides through the image pixels at a given stride and is able to extract meaningful features from the input image. A classic example is a 'beak detector' in order to help classify the image as a bird. Because these filters focus only on local pixels within the image, individually they gather much more meaningful information than a normal neural network would. In our case the NN is BART and the CNN is the Longformer. The same logic of the kernels applies to the sliding window attention in Longformer. As the authors explain, because this is a local attention mechanism, 'Given a fixed window size w, each token attends to $\frac{1}{2}w$ tokens on each side.' This in turn results in a computational complexity of $O(n \times w)$ which is clearly linear in terms of n. The figure below from the original paper gives a clear view of the advantage the sliding window attention.



The authors later mention the idea of a 'Dilated Sliding Window' and specifically tie it to dilated CNNs, but we won't review this any further because we focus on the global attention and the LED variant specifically. We argued that local attention was a smart solution, but it's not enough. You see, local attention is great for the granularities, but what happens if by the time we reach point B, the model has forgotten the details it dug up in point A? We need a mechanism that acts as a memory storage, a buffer for these details in order to store them securely. That's what global attention does. It chooses tokens and assigns them a 'Global' label, whereby the token attends to the whole document and vice-versa. This idea enables the model to keep track of the learned details while still viewing the document as a whole.

This combination of local + global attention is exactly the one the authors propose for LED. They replace the traditional full self-attention found in BART with a mechanism which allows input sequence up to 16k tokens in length. For the decoder, they instead opt to keep the architecture as is, arguing that summaries are relatively short even for long documents and they need full self attention to keep a bird's eye view of the document at all cases. And while newer architectures always improve on the flaws of their predecessors, they possess flaws of their own. With LED, it's a subtle one. As we mentioned earlier, LED incorporates the local + global attention in the encoder, but for global attention the global labels need to be set manually. This is inefficient. Moreover, cramming all the information gathered from 16k tokens into **one singular token** is risky to say the least.

That's why we look to the LongT5 [10]. It's based on the standard T5 model, but developed with specific attention mechanisms to handle the 16k tokens. The T5 is just like BART, a very good baseline for sequences up to 1k tokens; anything over that and it starts crumbling. Funnily enough, the attention mechanism in LongT5 offers a solution not only to its predecessor (the T5), but also to a much more competent model which already handles 16k tokens (LED). It implements Transient Global (TGlobal) attention, which basically works by dividing the input sequence into blocks. The global tokens are then automatically computed for each block, and all local tokens attend to these global representatives and vice-versa. The difference is that instead of having a manually calculated representative to store the compressed information, we now have a network of memories. Going back to our analogy of going from point A to B, TGlobal attention works as follows: tokens in point B don't have to hope they're important enough to be labeled 'global', all the info gathered is stored in a designated representative of the block, which can access any other representative of any other block.

Equipping the T5 (known for its strength in summarization tasks due to its "Text-to-Text" framework) with the scalability to process texts upwards of 16k tokens proved to achieved state-of-the-art results in metrics like Rouge-1, Rouge-2 and Rouge-L in datasets like arXiv and PubMed for generative tasks of text summarization. The results of the paper by the guys over at Google Research are shown for the aforementioned datasets. It is worth mentioning that we've decided not to include runs for the LongT5 for sequences longer than 4k tokens. However, this comes from limited compute reasons and not because it can't handle such long sequences.

Approach	arXiv R-1	R-2	R-L
DANCER PEGASUS	45.01	17.6	40.56
BigBird-PEGASUS (large)	46.63	19.02	41.77
HAT-BART	46.68	19.07	42.17
LED (large)	46.63	19.62	41.83
PRIMER	47.6	20.8	42.6
LongT5 (large - 16k input)	48.28	21.63	44.11
LongT5 (xl - 16k input)	**48.35**	**21.92**	**44.27**

Approach	PubMed R-1	R-2	R-L
DANCER PEGASUS	46.34	19.97	42.42
BigBird-PEGASUS (large)	46.32	20.65	42.33
HAT-BART	48.36	21.43	37.00
LongT5 (large - 16k input)	49.98	24.69	46.46
LongT5 (xl - 16k input)	**50.23**	**24.76**	**46.67**

5.3 Domain-Specific Challenges: The Legal Frontier

This subsection diverges slightly from the structure in relation to the prior ones. Here, rather than doing a thorough analysis of each model and how it builds on previous work, we look at the main difficulty of our task. Why is it so hard to deal with legal text and what do we need to change with our models to make them better at understanding legalese?

It's common knowledge that legal text is hard to understand. According to a group of cognitive scientists led by MIT professor Edward Gibson, it is understood that the reason for this style of writing comes from the need to convey a sense of authority. [11] Besides this, the authors believe that this kind of text has the tendency to create layers of depth, especially through definitions and explanations, which make the text much longer and difficult to keep track of. This, coupled with the fact that the human attention span is now shorter than that of a goldfish [12], makes for a hard time comprehending legal text. But if we can't understand, how can machines? As mentioned earlier, CNNs are great for tasks involving images. And just like humans being 'wired' to distinguish a dog from a cat, CNNs are fed a plethora of examples showing them which images correspond to dogs or cats. Subsequently, they learn from these examples and are able to classify unseen pictures correctly.

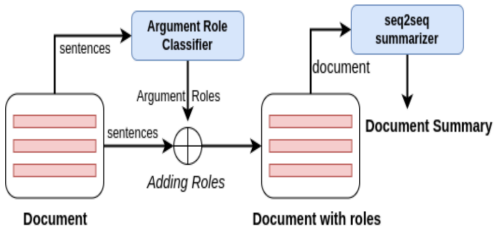
A similar approach should therefore be employed for legal text too. For instance, models like original BART and T5 were trained on plain English text, mainly news and articles. Consequently, they were poor to summarize long, dense legal documents. Even though these newer models are much more adept at storing long text, they still have to deal with the syntax and jargon of legalese. Thus, it becomes necessary to utilize domain-specific datasets that provide the high-quality human references required for a model to learn the precise logic of legislative and judicial language.

The first significant step in this frontier is the Billsum paper by Kornilova & Eidelman. [13] The paper introduced the first proper source of data available for tasks like summarization of legislation. The authors say they released the dataset 'to encourage research into automatic legislative summarization', seeing as up to that point, 'automatic summarization methods have been studied on a variety of domains, including news and scientific articles.' You see, for domains like news, the main information is almost always at the very beginning. On the other hand, you might find crucial clauses and provisions at any point in a legal text. Such differences call for a dataset specifically designed for this type of task, and Billsum provides exactly that.

Beyond identifying the structural difference between news and law, the authors also explore the challenges of cross-jurisdictional generalization. They tested whether models trained

on federal legislation could effectively summarize state-level bills. This contribution is vital because it suggests that a model's success in the legal frontier depends not just on its architecture, but on its ability to transfer linguistic patterns across different legislative frameworks. They concluded that it is in fact possible to train on US bills and generate summaries for California state bills. This, as Kornilova & Eidelman point out, allows summarization techniques to be applied reliably even for cases without a human written summary as a reference.

While BillSum addresses the structural challenges of legislative rules, a second direction in this frontier aims to focus on the argumentative nature of judicial decisions. A legislative bill (like in BillSum) prescribes what the law is. On the other hand, a judicial opinion is a discursive text that explains how the law is applied. To visit the other side of the coin, we shift our focus to the ArgLegalSumm paper by Elaraby & Litman. [14] The authors address the fact that standard sequence-to-sequence models are functionally blind to the argumentative anatomy of a court opinion, causing them to struggle with capturing core legal reasoning. To solve this, they integrate sentence-level argument mining — classifying clauses into legal roles like Issues, Reasons, and Conclusions — directly into the abstractive summarization workflow. The figure below illustrates how the approach works.



To analyze exactly how much these rhetorical cues help the model, Elaraby & Litman structure their evaluation around two distinct testing frameworks, labeled in their results as Oracle and Predicted. This division is crucial for assessing both the theoretical value and the practical viability of their approach:

- The Oracle Setting: In this scenario, the summarizer is given access to perfect, human-annotated argument roles. This serves as the experimental ceiling or upper bound. It answers a fundamental research question: If our model had flawless knowledge of the document's argumentative layout, how much better would the final summary be?
- The Predicted Setting: Since real-world documents do not come with pre-labeled human annotations, this setup evaluates a fully automated, end-to-end pipeline. Here, an independent classification model is first used to predict the argument roles of each sentence. The authors experimented with various models and found that LegalBERT yielded the highest performance, achieving a macro F1-score of 63.4% on the role classification task. These machine-generated labels are then blended into the text as special tokens before being processed by the abstractive summarizer. These results further support the claim that models need domain-specific knowledge in order to achieve better results.

The resulting metrics tell a highly informative story. While the Oracle configuration pre-

dictably secures the highest ROUGE scores, the Predicted configuration retains a significant portion of those gains, consistently outperforming strong, vanilla baseline architectures. This distinction is incredibly relevant to our task because it proves that even when accounting for the natural margins of error introduced by automated classification models like LegalBERT, explicitly guiding a transformer through the logic of an argument yields a far superior and legally coherent summary. It's useful to note the authors' description of markers in the study:
'We use 2 markers to denote the use of binary special tokens (i.e; <IRC>,</IRC>) and 6 markers to refer to the full set of argument role tokens. We include two markers sets to examine whether it's necessary to include explicit argument roles or if it's sufficient to highlight argumentative text only.' Below is a comprehensive view of the results obtained by Elaraby & Litman.

Setting	Experiment	Model	Rouge-1	Rouge-2	Rouge-L
		Unsupervised Extractive BERT	37.71	14.77	36.41
		Vanilla BART-Large	47.93	22.34	44.74
	Baselines	Vanilla LED-base	49.56	22.75	46.48
	arg-BART-Large	BART-Large + 2 markers	47.11	21.77	43.12
Oracle		BART-Large + 6 markers	46.80	22.14	44.14
		LED-base + 2 markers	49.64	**26.81**	46.70
	arg-LED-base	LED-base + 6 markers	**50.53**	26.31	**47.90**
		LED-base + 2 markers	49.50	*26.46*	46.60
Predicted	**arg-LED-base**	LED-base + 6 markers	*50.23*	26.29	*47.49*

Table 2: Summarization results on the test set. Best results **bolded**. Best results using predicted roles *italicized*.

Model	Rouge-1	Rouge-2	Rouge-L	Mean Summary Length
Vanilla LED-base	48.25	21.60	44.88	267
arg-LED-base + 2 markers	50.43	**27.76**	47.05	156
arg-LED-base + 6 markers	**50.73**	27.29	**47.30**	174

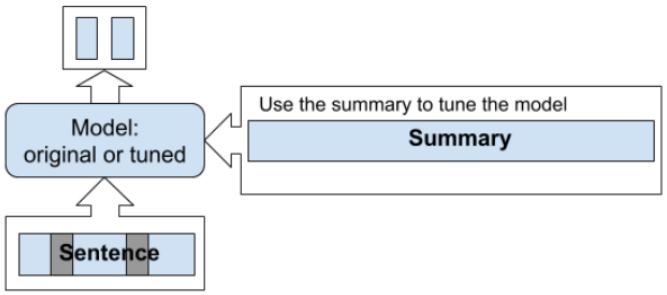
Table 3: Comparing Longformer (LED) summaries with sentences labeled as argumentative in reference summary.

5.4 Beyond ROUGE's Lexical Overlap: Functional Evaluation via BLANC

While traditional metrics like ROUGE remain the industry standard for measuring surface-level lexical overlap between machine-generated summaries and human reference texts, they can be inherently limited when evaluating purely abstractive models. In specialized fields like the legal domain, a language model might accurately compress or paraphrase a highly dense clause using simplified vocabulary. A strict n-gram matching metric like ROUGE may heavily penalize this variation, failing to recognize that the core legal intent has been fully preserved. To capture this missing dimension of semantic utility, this thesis incorporates the evaluation framework introduced in the Vasilyev et al. BLANC paper (Fill in the BLANC: Human-free quality estimation of document summaries). [15]

Rather than matching tokens against a gold-standard human reference, BLANC treats summary evaluation as a measure of functional information yield. The core mechanism utilizes an independent, pre-trained language model (such as BERT) to perform a masked token prediction task — also known as a Cloze task — on the sentences of the original source document. The language model attempts to reconstruct these hidden tokens under two distinct conditions: once in isolation using a blank filler, and once with the generated summary provided as predictive context. The final BLANC score is calculated by observing the precise accuracy boost the summary delivers to the model's performance. The authors present two primary variants of this framework: BLANC-tune, which fine-tunes the underlying model on the summary

text before text reconstruction, and BLANC-help, which directly concatenates the summary text to the document sentences during inference. For the implementation in this work, the BLANC-help variant is specifically utilized to quantify the immediate utility of our generated legal summaries. The diagram below displays the BLANC-help mechanism plus a side note from the authors on how it works:



Incorporating BLANC alongside traditional metrics allows our evaluation framework to benefit from assessing the summary directly against its original source document. This dual-metric approach creates a more robust, multi-dimensional evaluation pipeline. While ROUGE continues to ensure that our models maintain vocabulary and stylistic alignment with the legal experts who authored the ground-truth references, BLANC guarantees that even when our models choose to paraphrase or heavily condense dense legalese, they are still successfully preserving the fundamental informational utility of the original text.

6 Methodology

To rigorously assess the performance of state-of-the-art transformer architectures on long-form legislative text, this research thesis establishes an empirical, comparative evaluation framework. This chapter details the core hypotheses, the operationalization of the dataset, and the tripartite evaluation architecture designed to capture both lexical accuracy and deep legal coherence.

6.1 Core Hypotheses

The experimental design is structured around testing three primary research hypotheses. These hypotheses isolate the distinct and compounding effects of architectural context scaling, domain-specific pre-training, and the alignment of automated evaluation metrics with expert human judgment. In what follows, we lay out each hypothesis with its statement and rationale.

- Hypothesis 1 (H_1): The Context Window Expansion Effect
  - Statement: Scaling the model's active input context window across a scaled progression of 1,024, 4,096, 8,192, and 16,384 tokens will result in a statistically significant, monotonic increase in summary completeness and information retention.
  - Rationale: Standard models like BART-Base suffer from a severe "truncation penalty" when processing legislative documents, as vital amendments or statutory clauses are often located in the latter sections of a bill. By expanding the attention field using LongT5 and LED, the network should capture long-range dependencies and preserve context that is otherwise discarded by baseline truncation, leading to demonstrably higher factual coverage in the output summaries.
- Hypothesis 2 (H_2): Domain-Specific Pre-training
  - Statement: Architectural context expansion alone is insufficient for high-fidelity legal text summarization. Combining an extended context window ($16,384$ tokens) with legal domain-adapted pre-training (Legal-LED) will yield statistically superior performance in legal precision and syntactic coherence compared to general-domain models of equivalent or larger scale (LongT5-Base).
  - Rationale: Legal language features highly specialized vocabulary, distinct syntactic structures, and precise semantic conditions. A model pre-trained strictly on general web corpora (such as LongT5) lacks the conceptual boundaries necessary to correctly interpret and summarize legal jargon. Thus, the synergy of a wide attention window and a domain-adapted vocabulary is hypothesized to outperform a wide attention window alone.
- Hypothesis 3 (H_3): Evaluation Metric Alignment
  - Statement: Traditional, n-gram-based lexical overlap metrics (ROUGE-1, ROUGE-2, and ROUGE-L) will demonstrate a weak correlation with human-graded evaluations of legal summary quality. Conversely, reference-free semantic-functional metrics (BLANC) and LLM-as-a-judge evaluations will demonstrate a significantly stronger, statistically positive correlation with human expert scores.

- Rationale: ROUGE scores reward exact word matches but are blind to factual consistency and legal accuracy; a summary can achieve a high ROUGE score while completely misrepresenting a legal obligation through minor wording shifts (e.g., swapping ”may” for ”shall”), or worse, by missing negators like ’not’. Because BLANC measures the functional information yield of a summary, and an AI agent judge can evaluate semantic consistency, these two modern methodologies are expected to align much more closely with the nuanced quality assessments of human evaluators.

6.2 Dataset Operationalization and Truncation Simulation

To evaluate the validity of our core hypotheses, the raw corpus must be systematically mapped to our experimental model pipeline. The primary objective of this operationalization is to isolate the impact of input sequence length on summarization quality while controlling for all other variables.

Rather than treating the dataset statically, our methodology models the input as a variable-ceiling pipeline. We process identical legislative bills through distinct architectural pathways, purposefully varying only the input capacity while holding the core targets constant. This conceptual framework consists of two main pillars:

- The Variable-Ceiling Input Simulation
  To simulate the real-world hardware and architectural limits that practitioners face, we establish four clear structural zones for input ingestion. This progression allows us to map the precise curve of information retention as the context window scales:
  - The Truncated Baseline Zone (1,024 Tokens): Represents the standard transformer input cap. The raw legislative text is strictly truncated, simulating the information loss typical of standard sequence-to-sequence models. This serves as our control environment to measure the baseline ”truncation penalty.”
  - The Intermediate Long Zone (4,096 Tokens): Introduces early-stage context expansion, capturing mid-length documents and letting us observe the initial marginal gains of moving beyond standard dense attention limits.
  - The Extended Long Zone (8,192 Tokens): Serves as a high-capacity intermediate threshold. Because a significant portion of detailed legislative bills fall within this length range, this zone acts as a critical benchmark for evaluating models that can handle substantial document structures without yet requiring ultra-long context configurations.
  - The Full-Context Zone (16,384 Tokens): Ingests the entire document unbroken, removing the input limit as an experimental bottleneck. This exposes the models to the complete legal narrative — including trailing amendments and final clauses — to test our primary hypothesis (H_1).
- Standardization of Output Constraints
  A major challenge in evaluating summarization models is ”length bias,” where models that write longer, wordier summaries are artificially rewarded by automated metrics simply because they have more opportunities to match target words. To isolate architectural

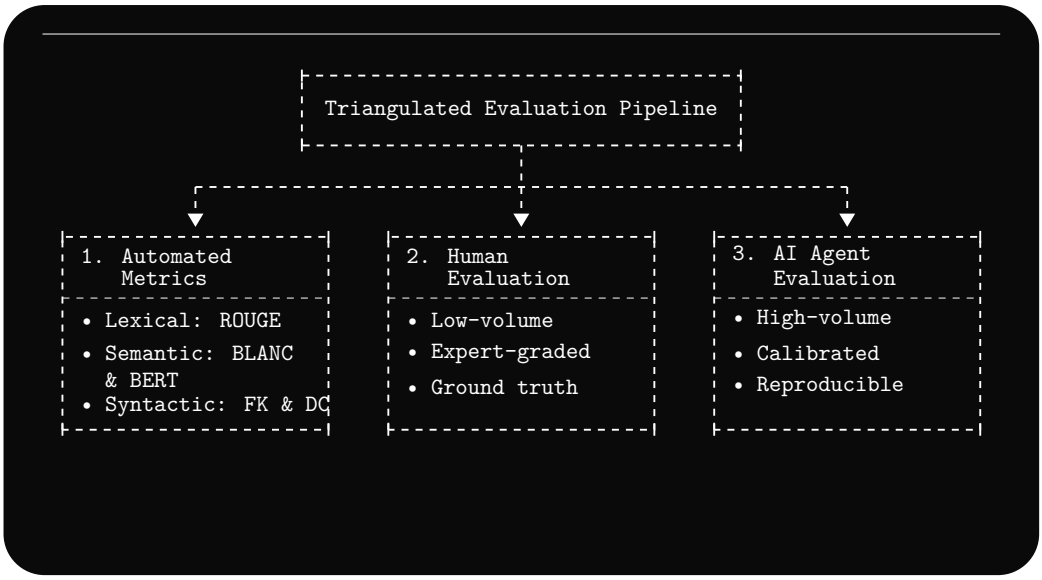
comprehension from sheer verbosity, our experimental setup enforces strict, standardized output ceilings across all models.

By forcing every network to compress its final summary into an identical, tightly controlled length range, we ensure that the models are evaluated solely on their ability to identify and prioritize the most legally critical information, rather than their tendency to generate runaway text. This establishes a level playing field, ensuring that any improvements in our metrics represent genuine gains in information synthesis and semantic precision.

6.3 The Triangulated Evaluation Pipeline

Evaluating automatic summarization in highly specialized domains like law is notoriously difficult. Traditional NLP evaluations rely almost exclusively on automated, n-gram-based metrics. However, in the legal domain, a summary can achieve a statistically perfect overlap score while introducing fatal factual errors, or conversely, be penalized for using correct legal synonyms not present in the reference text.

To overcome these biases, this research implements a Triangulated Evaluation Pipeline. Rather than relying on a single source of truth, we validate our model outputs across three distinct, overlapping dimensions: automated lexical and functional metrics, qualitative human grading, and synthetic AI agent evaluations. This multi-perspective design ensures that our conclusions are anchored by both mathematical reproducibility and human expertise.



6.3.1 Pillar 1: Automated Computational Metrics (Lexical vs. Functional)

Our automated evaluation splits into three distinct paradigms to capture the surface-level structure and the semantic utility of the summaries, while also focusing on readability and

syntactic coherence:

- Lexical Overlap (ROUGE-1, ROUGE-2, ROUGE-L): We calculate traditional ROUGE metrics to establish a historical baseline. These metrics measure the strict mathematical overlap of unigrams, bigrams, and the longest common subsequences between the generated summary and the gold-standard reference. While useful for tracking syntactic similarity, they act purely as a proxy for surface-level writing patterns.
- Functional Information Yield (BLANC, BERT): To measure if a summary actually helps a reader understand the source document, we implement BLANC (a reference-free, cloze-task-based metric). Instead of comparing the summary to a human-written text, BLANC uses the summary to help a pre-trained language model predict masked words in the original, full-length document. If the summary is highly informative, the model's prediction accuracy increases, providing a functional measure of semantic information retention.By utilizing token-level contextual embeddings, BERTScore computes a soft-alignment semantic similarity matrix between the generated summary and the reference text. This isolates true semantic equivalence, ensuring that models are rewarded for accurate legal paraphrasing even when utilizing varied vocabulary or alternative phrasing.
- Syntactic Complexity and Readability (Flesch-Kincaid, Dale-Chall): A critical, often overlooked objective of machine-driven legal summarization is the democratization of dense information. Legal texts are notoriously insulated by "legalese" — convoluted sentence structures and low-frequency vocabulary that elevate cognitive load. Therefore, our framework establishes a dedicated readability matrix to track whether models successfully translate dense statutory prose into accessible briefs. The FK metric targets syntactic complexity by analyzing the mathematical ratio of sentence length to syllable density. It establishes the approximate institutional education level required to comprehend the summary, tracking whether a model is generating concise summaries or runaway, hyper-complex sentences. Meanwhile, the DC metric assesses semantic difficulty by calculating the ratio of "difficult words" against a baseline vocabulary of familiar language. It provides an exact diagnostic for tracking how effectively a model strips out impenetrable legal jargon or, conversely, whether an expanded context window causes a model to simply copy obfuscated terminology directly into the output.

6.3.2 Pillar 2: Qualitative Human Evaluation (The Empirical Anchor)

While automated metrics provide scale, human eyes remain the absolute gold standard for measuring real-world utility. We execute a targeted human evaluation study where human annotators grade a randomized subset of model outputs.

Evaluators score the summaries blindly (with no knowledge of which model generated which text) on a standard 1–5 Likert scale across three critical qualitative dimensions:

- Faithfulness / Accuracy:
  Does the summary match the evidence and avoid incorrect or unsupported claims?
- Clarity:
  Is the summary easy to read and understand?

- Usefulness:
  Would this summary help someone quickly understand the key points?

We managed to track 75 responses across 15 different documents, which means each document was rated five times.

6.3.3 Pillar 3: AI-as-a-Judge Evaluation (The Synthetic Scaler)

A persistent challenge in qualitative evaluation is participant fatigue, which limits the volume of summaries that human judges can realistically grade. To bridge the gap between low-volume human evaluation and high-volume automated metrics, this research introduces a calibrated, multi-agent synthetic evaluation layer. By employing advanced Large Language Models (LLMs) as independent evaluator agents, we scale our qualitative assessment across a broader sample of the dataset while maintaining strict alignment with human evaluators. [16]

This synthetic evaluation cohort is designed around two distinct, state-of-the-art LLM judges: Claude Sonnet 4.6 and Gemini 3.1 Flash-Lite. These models represent two different tiers of LLM architecture — a highly sophisticated, frontier-class model (Claude) and a highly efficient, speed-optimized model (Gemini) — allowing us to cross-validate evaluation behavior and ensure our synthetic scores are not biased by a single model's architectural quirks. The scale and mechanical execution of this synthetic evaluation pipeline are structured as follows:

- Evaluation Volume: The evaluation set comprises 15 complex legislative documents. Feeding these documents through our four primary models (BART, LED-Base, LED-Large, and LongT5) yields a pool of 60 generated candidate summaries.
- Dual-Judge Replication: Every individual summary is evaluated independently by both Claude Sonnet 4.6 and Gemini 3.1 Flash-Lite. To control for the inherent stochasticity (randomness) of LLM outputs, each judge executes the evaluation script twice using entirely different random seeds. This produces 4 distinct output files containing a total of 240 structured ratings.
- Double-Blind Shuffling Protocol: LLM judges are highly susceptible to positional bias (the tendency to favor options presented first in a prompt) and signature bias (the ability to recognize and favor a specific model's writing style). To eliminate these confounders, we implement a rigorous, seed-controlled blinding process. Before a judge receives the summaries for a given legislative bill, the four candidate summaries are shuffled into randomized positions and anonymized with neutral identifiers (e.g., Summary A, Summary B, Summary C, Summary D).

The second evaluation run is not a different methodology; rather, it is a direct replication utilizing a completely unique seed to reshuffle the summary sequence. By keeping the models blind to the identity of the generator and varying the sequence presentation, we guarantee that any consistency in scoring reflects genuine semantic and legal quality.

7 Data

7.1 Dataset Selection & Rationale

For the core evaluation of long-document legal summarization, this thesis utilizes the BillSum dataset. The primary rationale for selecting this specific corpus is its structural composition: it provides the complete, unmitigated text of legislative bills alongside high-quality, human-authored ”ground truth” summaries. These reference summaries serve as the definitive gold standard, allowing for a quantitative evaluation of an abstractive system’s capacity to retain critical legal provisions without omitting vital context.

The primary corpus consists entirely of legislative documents. Text of this nature presents unique structural hurdles for standard language processing pipelines, characterized by dense technical jargon, deeply nested sub-sections, and a total absence of the typical ”lead bias” found in standard news benchmarks.

7.2 Partitioning and Data Filtering

The official BillSum dataset is natively divided into three distinct data partitions:

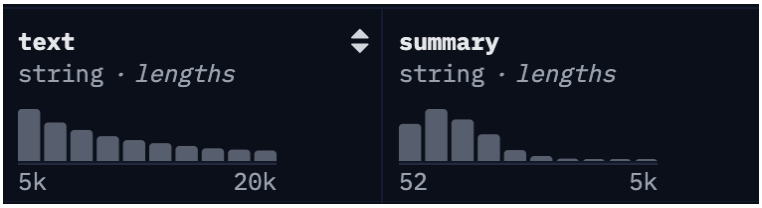
- train: The primary training set composed of US Federal bills collected from the United States Government Publishing Office (GPO), spanning the 103rd to 115th sessions of Congress. This partition comprises 18.9k data points.
- test: The standard testing partition composed of US Federal bills from the same Congressional sessions. The test set has 3.27k data points.
- ca_test: A specialized testing partition focusing on state-level legislation scraped from the California legislature’s website during the 2015–2016 session. This partition only includes 1.24k data points.

For the purposes of this experimental framework, the ca_test split was entirely disregarded. This filtering decision was made due to two limiting factors. First, the California test partition contains a highly restricted number of samples compared to the federal data. Second, the dataset lacks a corresponding state-level training partition (ca_train) to support it. Relying on a test set without a matching domain-specific training framework introduces unaddressed cross-jurisdictional generalization variables that fall outside the scope of our core baseline comparison. Consequently, this study restricts its data pipeline entirely to the US Congressional partitions.

7.3 Text Length and Statistical Characteristics

A critical property of the BillSum dataset is its deliberate focus on mid-length and long-form legislation. The dataset creators filtered the raw scraping results to explicitly capture documents ranging from 5,000 to 20,000 characters in length. This range was selected because exceptionally short bills typically introduce minor, single-sentence statutory changes that do not require a separate summary, while ultra-long documents are frequently massive compilations of completely unrelated sections.

For the target references, the human summaries are subject to a strict 2,000-character upper limit, with around 90% of the summaries falling comfortably within this threshold. Interestingly, statistical analysis of the corpus reveals little to no direct correlation between the length of a bill and the length of its corresponding human summary. A highly dense 18,000-character bill may be compressed into a clean 1,200-character summary, presenting a high-compression, abstractive challenge where a model must extract systemic legal intent rather than relying on structural extraction.



7.4 Data Tokenization and Truncation Constraints

For the technical execution of the experimental pipeline, the raw text fields provided by the BillSum dataset are integrated directly into the model training layers without manual, string-level heuristic interventions (such as regular-expression text cleaning). Instead, the data preparation pipeline focuses purely on transforming the raw textual strings into structured, numerical token streams via model-specific tokenizers. The preprocessing pipeline implements a unified mapping function that acts on both the source legislation and the reference targets simultaneously. The processing sequence executes as follows:

- Source Tokenization: The raw text of the bill (examples[”text”]) is mapped to input IDs using the model’s native tokenizer, capped at a designated max_input_length.
- Target Tokenization: The human reference summary (examples[”summary”]) is processed similarly, capped at a designated max_target_length.
- Label Alignment: The generated input IDs of the target summary are explicitly assigned to the labels key within the data dictionary (inputs[”labels”] = targets[”input_ids”]), structuring the data for sequence-to-sequence loss calculation.

Crucially, the pipeline enforces a strict truncation policy (truncation=True) during tokenization. This algorithmic constraint is particularly significant for baseline architectures like BART, which feature a rigid context window limit of 1024 tokens. For documents whose sequence length exceeds these predefined configurations, the trailing tokens are automatically discarded. Once tokenized, the original text and summary fields are removed from the active dataset rows (remove_columns=[”text”, ”summary”]), leaving a data structure optimized for parallelized batch processing.

8 Models

8.1 Model Selection Strategy

The selection of the language models for this thesis is guided by a systematic benchmarking strategy designed to test two critical factors in automated legal summarization: the absolute length of the input context window and the impact of domain-specific pre-training. Rather than arbitrarily evaluating networks, a controlled progression of models was chosen to establish a clear performance trajectory as architectural constraints are incrementally removed.

The primary target sequences generated by the models are subject to a uniform constraint across all architectures, with a maximum generation limit set to 256 tokens. This ceiling ensures architectural consistency during inference and prevents generation length variations from artificially skewing evaluation metrics, directly mirroring the typical distribution of the BillSum reference targets.

Our models span four specific variants, ranging from tight, general-purpose baselines to highly expanded, domain-adapted networks. Their exact Hugging Face identifiers and sequence parameters used are detailed in the table below: [17]

Model	**Name**	**Identifier**	**Max Input**	**Config Max Input**	**Config Max Output**	**Pre-training Domain**
BART base	facebook/bart-base	Full Dense	1024	1024	256	General Web Corpus
LongT5 base	google/long-t5-tglobal-base	Transient Global	16384	4096	256	General Web Corpus
Legal LED base	nsi319/legal-led-base-16384	Local + Global	16384	8192	256	Legal Adaptations
Legal LED large	0-hero/led-large-legal-summary	Local + Global	16384	16384	256	Legal Adaptations

8.2 Architectural Selection Rationale

8.2.1 Establishing the Low-Context Baseline: BART Base

The BART base architecture serves as the definitive structural baseline for this study. Utilizing standard, full quadratic attention, it represents the traditional generation of sequence-to-sequence transformers constrained by a rigid 1,024-token input window. Because the data analysis in Chapter 7 established that the vast majority of raw US Congressional bills sig-

nificantly exceed 1,024 tokens, this model forces aggressive truncation on the data pipeline. Including this baseline allows us to measure the exact performance deficit caused by discarding the trailing text of a bill, confirming whether advanced long-context models are practically necessary.

8.2.2 Evaluating General Long-Context Scale: LongT5 Base

The LongT5 base model was selected to bridge the gap between restricted baselines and ultra-long context windows. By employing a specialized Transient Global attention mechanism, it scales the input capacity up to 4,096 tokens without experiencing the catastrophic memory explosions typical of standard quadratic attention. Crucially, LongT5 is trained entirely on general web text (the C4 corpus). By introducing a model that can handle substantial text lengths but lacks domain-specific legal training, our experimental matrix can isolate whether raw sequence capacity alone is sufficient to resolve legal document complexity.

8.2.3 Evaluating Specialized Domain Scale: Legal LED Base & Large

To test the apex of domain-specific adaptation combined with extreme context windows, the Longformer Encoder-Decoder (LED) architecture was selected in both its base and large variants. Both models swap quadratic attention for an efficient sparse attention mechanism, allowing us to utilize input limits up to 8,096 tokens for the base model and an expansive 16,384 tokens for the large model.

Unlike their general-purpose counterparts, both checkpoints have undergone targeted pretraining and secondary fine-tuning on specialized legal corpora. The inclusion of these models allows the experimental framework to determine the precise compounding benefit achieved when an architecture is simultaneously granted the capacity to read an entire legislative document unbroken and the vocabulary initialization required to parse dense legalese.

9 Experiments

9.1 Model Fine-Tuning and Execution (What We Did and Why)

9.1.1 Compute Infrastructure and Resource Constraints

To balance access to high-performance hardware with strict computational resource constraints, all fine-tuning and inference pipelines were executed within the Lightning AI cloud ecosystem.[18] This setup utilized a cloud-managed workspace instance backed by a single NVIDIA H200 GPU. Leveraging Lightning AI's monthly credit allocation allowed for the cost-effective deployment of frontier-grade hardware capable of handling deep memory states. All modeling workflows were implemented natively in PyTorch, ensuring maximum control over data tensors and custom pipeline mechanics.

9.1.2 Model Specifications and Architectural Configurations

Four primary models were selected from the Hugging Face repository to construct our context-scaling ladder. Each architecture was intentionally capped at explicit input ceilings to isolate the operational impact of text truncation on legal text processing:

- BART_base (facebook/bart-base): Serving as our structural control baseline, this model was operated at its absolute architectural ceiling of $1,024$ input tokens.
- LongT5_base (google/long-t5-tglobal-base): Chosen to evaluate general-domain sparse-attention mechanisms. While capable of ingesting ultra-long sequences, it was intentionally capped at a maximum of $4,096$ input tokens to evaluate early-stage context expansion performance.
- Legal_LED_base (nsi319/legal-led-base-16384): Employed to analyze the initial compounding effects of base-scale domain adaptation. This network was capped at an intermediate high threshold of $8,192$ tokens.
- Legal_LED_large (0-hero/led-large-legal-summary): Deployed as our primary high-capacity model, this architecture was run at its maximum operational ceiling of $16,384$ input tokens to parse completely unbroken statutory texts.

9.1.3 The Core Experimental Matrix

To systematically map performance variations across different input and output configurations, a total of 34 discrete experimental runs was executed. Every successful model configuration was trained uniformly for 3 epochs using the AdamW optimizer with a static learning rate of 5×10^{-5}.

The fine-tuning runs were executed across an incremental ladder of input sizes (512, $1,024$, $2,048$, $4,096$, $8,192$, and $16,384$ tokens) paired with two distinct target summary lengths (128 and 256 max tokens). Models were systematically decommissioned and removed from successive runs the moment the experimental text requirements exceeded their specific input caps. This is best understood from the diagram below.

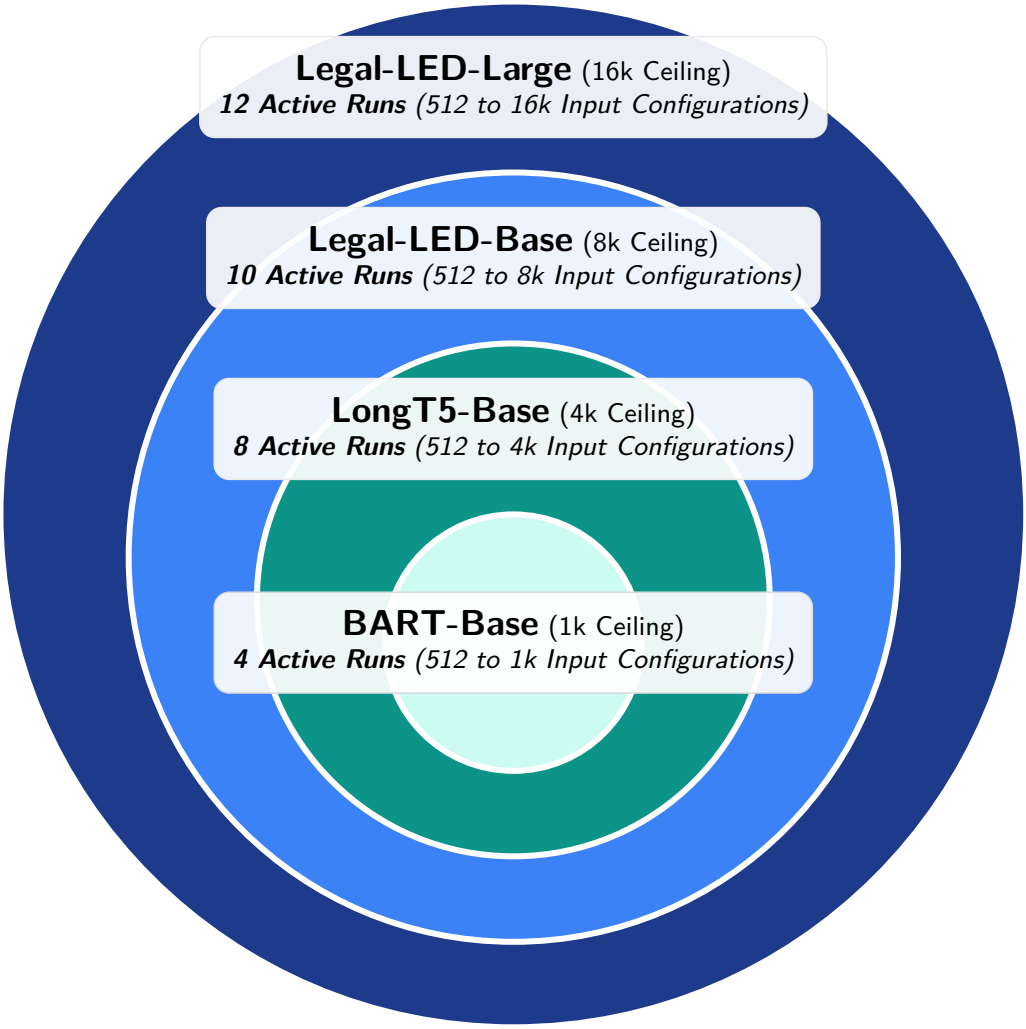


Figure 1: Euler Inclusion Diagram of the Experimental Run Allocation. The model configurations are nested hierarchically by architectural input capacity, tracking the active model fleet across the 34 total fine-tuning runs.

9.1.4 Data Pipeline and Memory Optimization Strategies

Processing extremely long sequences introduces massive memory and data transfer overhead. While lower-to-mid-token length configurations (512 to $4,096$ inputs) were highly stable at a default batch size of 32, scaling into ultra-long territories ($8,192$ and $16,384$ inputs) required a specialized memory and pipeline optimization strategy to run successfully on a single H200 GPU.To prevent out-of-memory errors and optimize hardware throughput during these

demanding high-capacity runs, the following engineering protocols were enforced:

- Batch & Gradient Management:
  The base hardware batch size was reduced from 32 down to 8. To preserve the effective training batch size and maintain gradient stability, we implemented gradient accumulation steps = 2.
- Precision Scaling:
  Native PyTorch bfloat16 mixed precision was activated, reducing memory footprints by storing activations in low-precision tensors while preserving high-precision stability for critical weights.
- Parallel Data Ingestion:
  To remove data bottlenecking between the CPU storage and GPU cores, the PyTorch DataLoader was configured with pin_memory=True (enabling accelerated page-locked memory copies), num_workers=4 (distributing background batch creation across 4 parallel CPU processes), and a prefetch_factor=2 to ensure continuous data streams to the H200 cores.
- Parallel Tokenization Preprocessing:
  During the initial dataset map and tokenization sequences, we initialized num_proc=4 to execute parallel multi-process text mapping across the raw BillSum dataset, significantly compressing preprocessing overhead before training began.

9.2 Evaluation Execution Protocol

9.2.1 Automated Quantitative Evaluation Metrics

The primary automated layer measured lexical precision and semantic information retention using two open-source frameworks:

- ROUGE Metrics: We deployed the standard rouge-score python package to calculate ROUGE-1, ROUGE-2, and ROUGE-L F1 scores. This quantified the unigram overlap, bigram overlap, and longest common subsequence matching between the generated summaries and the gold-standard BillSum reference summaries.
- BLANC Metric: To evaluate information preservation independent of literal wording overlaps, we cloned the official open-source repository from PrimerAI. The BLANC evaluation pipeline was executed using the pre-trained bert-base-uncased transformer architecture to run token-masking and document reconstruction tasks, measuring how effectively each summary aided a language model in understanding the original source text.
- BERT F1 Score: To quantify deep semantic similarity and capture accurate legal paraphrasing, we utilized the official bert-score Python library. This pipeline calculated token-level contextual embeddings across a dense similarity matrix to generate an overall semantic F1 score. By measuring the structural meaning vectors rather than surface character tokens, this metric isolated instances where models preserved legal concepts through alternative phrasing, preventing the artificial penalties that rigid string-matching tools frequently impose on abstractive architectures.

- Readability and Syntactic Complexity Metrics: To evaluate whether our context-scaling models successfully democratized dense legislative prose or inadvertently preserved impenetrable "legalese," we integrated the open-source textstat Python library. Every generated summary was programmatically evaluated across two classic readability formulas:
  - Flesch-Kincaid Grade Level (FK): This metric targets syntactic complexity by computing a linear combination of average sentence length and average syllable density per word. Higher scores indicate longer, more complex sentence structures. Within a legal domain context, tracking fluctuations in this score exposes whether a model is generating coherent, human-digestible prose or falling into patterns of runaway, unpunctuated clause copying.
  - Dale-Chall Readability Score (DC): Unlike formulas relying strictly on syllable or character counts, this metric assesses semantic difficulty by calculating the exact percentage of "difficult words" — defined as tokens failing to appear on a standardized baseline list of approximately 3,000 familiar words. A higher score signifies heavy reliance on low-frequency vocabulary and technical terminology. This provided a precise diagnostic tool for tracking whether expanding a model's input context window causes it to default to copying obfuscated statutory jargon directly into the output.

9.2.2 Human Expert Evaluation Framework

To gather high-fidelity qualitative data and validate the automated metrics, a targeted human evaluation trial was conducted using professional legal evaluators.

- The Evaluator Cohort: The evaluation panel comprised 75 human participants with specialized legal backgrounds. The overwhelming majority (67 individuals) were law graduate students. To ensure a balance of academic authority and practical industry experience, the cohort also included practicing attorneys, law school professors, and corporate compliance experts with formal legal training.
- Sampling and Survey Architecture: Reading multi-page legal bills alongside four comparative summaries introduces severe cognitive fatigue, which can compromise annotation quality. To mitigate this risk, we implemented a sparse sampling matrix. We randomly selected 15 benchmark documents from our test outputs. The survey was constructed on the Qualtrics platform using a strict "one document per survey instance" layout.
- Blinding and Distribution Math: Each evaluator was presented with a single source excerpt followed by four generated summaries labeled blindly as Summary A, B, C, and D. The underlying model identities were hidden from the participants. This distributed design resulted in exactly 5 independent human reviews per legal document (15 documents $\times$ 5 responses = 75 total survey instances).
- The Grading Scale: For each summary, raters answered three targeted questions evaluating Faithfulness/Accuracy, Clarity, and Usefulness. Every dimension was scored on a uniform $1-5$ Likert scale, where 1 represented "very poor," 3 represented "acceptable," and 5 represented "excellent."

9.2.3 Synthetic LLM Evaluation Pipeline (The AI Judges)

To explore the viability of automated qualitative auditing, we established a synthetic evaluation pipeline that mirrored our human trial design exactly. The evaluation suite leveraged direct API integrations with two frontier models acting as independent synthetic judges: Claude 4.6 and Gemini 3.1 Flash-Lite.

- API Configuration: To guarantee deterministic, reproducible grading outputs and minimize stochastic variance, the sampling temperature for all API connection requests was set to a rigid $T = 0.0$.
- The Structural Prompt Template: Both models received identical instructions. The prompt explicitly framed the AI as an automated legal judge executing a structured survey rubric, forcing it to look only at the enclosed evidence text and output scores via a standardized JSON block. The exact prompt string used in our execution scripts is provided in the appendix.
- Mitigation of Positional Bias: Large Language Models are highly susceptible to positional bias, frequently awarding higher scores to options placed earlier in the prompt context. To isolate and neutralize this artifact, each synthetic judge executed two independent evaluation runs per document using a label-shuffling scheme. For instance, if Run 1 mapped the summaries as A $=$ BART, B $=$ Legal-LED-Large, C $=$ LongT5 and D $=$ Legal-LED-Base, Run 2 systematically made sure the assignment differed so a possibility would be A $=$ LongT5, B $=$ Legal-LED-Base, C $=$ Legal-LED-Large and D $=$ BART.

Initial inspection of the evaluation logs revealed shifts in the assigned scores across the two runs, verifying that the permutation script successfully decoupled model quality from text placement. The final synthetic score for each model was calculated as the mean across both shuffled runs.

9.3 Brief Chapter Summary

This chapter detailed the execution of 34 model training runs on an NVIDIA H200, optimized via mixed-precision and gradient accumulation to handle up to 16k context windows. We established a three-tiered evaluation pipeline comprising lexical metrics (ROUGE/BLANC), a blinded survey distributed to 75 legal experts, and a double-run shuffled synthetic judge framework to eliminate positional bias. With these experimental configurations and evaluation protocols successfully set up, Chapter 10 presents the comparative results across all automated, human, and AI-driven benchmarks.

10 Results

10.1 Quantitative Performance: Automated Metrics

To establish a baseline evaluation of string alignment, semantic reconstruction, and linguistic complexity across our context-scaling ladder, we isolate the peak-performing configuration for each of our four primary model architectures. The exhaustive 34-run evaluation matrix tracking every tested context and generation configuration is documented for full reference in the Appendix. The table below isolates the best performing configuration for each model, providing a clear snapshot of the relative performance across the full spectrum of automated metrics.

Table 2: Peak Quantitative Performance by Model Architecture. Values represent the highest achieved metrics for each model class across all context and target generation configurations.

Model Class	Config	FK	DC	R1	R2	RL	BERT	BLANC
BART_base	1024/256	**42.44**	**15.22**	0.46	0.26	0.34	0.83	0.17
LongT5_base	4096/256	66.34	18.13	0.53	0.34	0.41	0.84	0.23
Legal_LED_base	8192/256	68.12	18.41	0.53	0.33	0.40	0.84	0.23
Legal_LED_large	16384/256	61.87	17.80	**0.56**	**0.35**	**0.42**	**0.85**	**0.24**

A systematic analysis of the data reveals three major trends regarding how scaling up context boundaries impacts automated summary metrics:

- The Context-Window Scaling Premium: There is a clear, monotonic positive correlation between an architecture's context ceiling and its downstream metric performance. The baseline BART-base model, bound by its rigid 1024-token limit, achieves a peak ROUGE-1 score of 0.46 and a BLANC score of 0.17.
  By expanding the receptive field to 4096 tokens via LongT5, ROUGE-1 improves significantly to 0.53.The ladder culminates with Legal-LED-large at a 16384-token context window, which secures the global dataset maximums across ROUGE-1 (0.56), ROUGE-L (0.42), and BERTScore (0.85). This confirms that truncating expansive legal documents to fit traditional model boundaries discards foundational, high-weight text elements necessary for high-fidelity summary construction.
- The Semantic vs. Lexical Paraphrasing Trade-off: While ROUGE metrics track strict character-to-character matching, the addition of BERTScore reveals a deeper look at semantic fidelity. Across the entire experimental matrix, BERTScore values remain tightly bounded between 0.80 and 0.85.
  The fact that Legal-LED-large achieves a 0.85 BERT F1 score alongside its peak ROUGE metrics indicates that its summaries are not merely matching strings, but are maintaining deep contextual vector alignment with the reference texts.The stable, high baseline of BERTScore suggests that even when models use slightly different phrasing or legal synonyms (which penalizes their ROUGE scores), they are successfully preserving the core legal intent and meaning across all token capacities above $1k$.

- The Runaway Complexity Penalty: While expanding the token budget improves alignment metrics, it extracts a severe penalty regarding linguistic accessibility. A macro-trend visible across all 34 runs in the appendix is the behavior of the readability scores when target generation lengths scale from 128 to 256 tokens. For instance, scaling Legal-LED-large from an input/output configuration of $8192/128$ to $8192/256$ drives ROUGE-1 up from 0.50 to 0.55, but causes its Flesch-Kincaid (FK) Grade Level to skyrocket from 43.44 to a massive 68.45. Similarly, the Dale-Chall (DC) scores climb well past 15.0 and 18.0, signaling an extreme concentration of low-frequency vocabulary.
  Because an FK score above 20.0 already signifies highly abstract academic text, scores in the $40–70$ range indicate a structural phenomenon: when abstractive models are given an expanded target token budget (256 tokens), they do not necessarily write more clear, cohesive paragraphs. Instead, they begin copying massive, unpunctuated clauses and cascading structural definitions directly from the source text.This introduces a clear optimization conflict: a larger target window yields superior text-retention metrics on paper (ROUGE/BLANC), but generates dense, runaway ”legalese” sentences that dramatically increase cognitive load for a human reader.

NB: Methodological Note on Global Optima: It is critical to clarify that the configurations displayed in the table represent a selection of architectural peaks chosen to isolate variables across our context ladder, rather than a presentation of absolute global maxima across all 34 runs. A granular audit of the full results table reveals two vital performance anomalies.
First, the absolute global maxima for both information preservation (BLANC = 0.25) and bigram precision (ROUGE-2 = 0.36) are actually achieved by the sub-maximal context window of Legal-LED-Large at the $8192/256$ configuration, outperforming its $16384/256$ counterpart.
Second, the absolute ”best” readability scores (the lowest linguistic complexity) belong to the initial, minimal context run of BART-base ($512/128$), which yields an FK of 34.40 and a DC of 14.21.This confirms that while the $16k$ window secures the highest unigram text retention (ROUGE-1), it does not automatically yield the global peak for every sub-metric, highlighting a delicate trade-off between absolute context capacity, semantic density, and output readability.

10.2 Human Evaluation Analysis

While quantitative readability metrics provide objective insight into syntactic complexity, they fail to capture the semantic integrity, communicative clarity, and practical utility of machine-generated legal text. To bridge this gap, a human evaluation study was conducted using a sample of legal professionals and advanced law students. Evaluators assessed the generated summaries across fifteen distinct legislative documents on a five-point Likert scale across three core dimensions: Faithfulness, Clarity, and Usefulness. This section presents the empirical findings of the human evaluation, analyzing the performance trade-offs between general-purpose language models and domain-specific architectures.

10.2.1 Global Macro-Performance and the Domain Pre-Training Premium

Aggregating the human evaluator scores across all fifteen test documents reveals a clear performance tier that validates the necessity of domain-specific model adaptation. The empirical macro-averages for each architecture are detailed below:

Model Class	Average Faithfulness	Average Clarity	Average Usefulness
BART_base (Baseline)	3.06	**4.34**	3.18
LongT5_base	3.32	3.69	3.16
Legal_LED_base	**3.67**	4.13	**3.71**
Legal_LED_large	3.35	4.31	3.55

The specialized Legal_LED_base architecture emerged as the top-performing model under human scrutiny, securing the highest macro-averages in both Faithfulness (3.67) and Usefulness (3.71). This finding demonstrates that pre-training on legal corpora equips the model with the structural and semantic framework necessary to capture complex legislative mandates accurately. Interestingly, scaling the architecture to Legal_LED_large yielded a minor decrease in macro-averages for Faithfulness (3.35) and Usefulness (3.55), while improving systemic Clarity (4.31). This suggests that while the larger parameter space enhances linguistic fluency and sentence coordination, it occasionally introduces slight over-generalizations that legal experts penalize for omitting hyper-specific details.

10.2.2 The Fluency vs Fidelity Trade-off

A profound finding of this human evaluation is the inverse relationship between superficial prose clarity and actual legal fidelity, exposing a critical vulnerability in general-purpose baselines. The standard BART_base baseline achieved the highest overall Clarity rating from human evaluators (4.34). However, this high fluency came at a direct cost to accuracy, as the model simultaneously recorded the lowest macro-average for Faithfulness (3.06).

When human legal experts evaluated BART_base summaries, they noted that the text read exceptionally well as standard English prose. However, because the general baseline optimizes for token economy and simple clause structures, it consistently stripped away vital conditional statements, jurisdictional limitations, and effective dates from the legislative text. In a professional legal environment, an easy-to-read summary that omits statutory exceptions is not only unhelpful, but potentially hazardous. The domain-trained Legal_LED architectures, by contrast, willingly sacrifice conversational simplicity to retain dense, multi-clause logic, ensuring that the critical core parameters of the law remain uncompromised.

10.2.3 Architectural Resilience in Long-Context Summarization

The evaluation highlighted significant divergence in model stability when processing long, highly complex statutory documents. A prime example is observed in the evaluation of Document 21026 (referring to the Protection from Sexual Predators Act), where the performance gap between long-context specialized architectures and standard configurations expanded dramatically:

- Legal_LED_large: Faithfulness: 5.00 | Clarity: 4.80 | Usefulness: 5.00
- Legal_LED_base: Faithfulness: 5.00 | Clarity: 4.40 | Usefulness: 4.80

- LongT5_base: Faithfulness: 2.80 | Clarity: 3.80 | Usefulness: 2.80
- BART_base: Faithfulness: 2.00 | Clarity: 4.20 | Usefulness: 2.00

For Document 21026, the BART_base baseline suffered a severe drop in performance, plunging to a score of 2.00 in both Faithfulness and Usefulness. This degradation stems directly from context-window exhaustion; constrained by its native input limits, the model was forced to truncate the vast majority of the source bill, rendering a summary that ignored late-stage clauses and subsequent amendments.

Conversely, Legal_LED_large achieved a perfect score of 5.00 for both Faithfulness and Usefulness on the same text. Backed by its sparse global-local attention mechanism, the model successfully ingested the entire statutory document, processed the long-range dependencies, and extracted a comprehensive synthesis that human evaluators deemed flawless for practical utility.

10.2.4 Empirical Exceptions and Outliers

While the superiority of domain pre-training holds true as a macro-trend, the granular document-level data reveals specific edge cases where standard models outperformed the specialized variations. In Document 10191 (referring to the Enhancement of Transition Assistance Program), BART_base led the cohort with a Faithfulness score of 4.60 and a Usefulness score of 4.20, outperforming Legal_LED_base (Faithfulness: 3.40, Usefulness: 3.40). Similarly, in Document 13903 (referring to the Helping Our Students Communicate Act), the generic LongT5_base achieved the highest Faithfulness score of 3.80, whereas Legal_LED_large dropped to a study-wide low of 1.80.

An analysis of these specific source bills reveals that they were brief, structurally linear resolutions that lacked nested statutory conditions or dense legal jargon. In these instances, the specialized architectures suffered from ”over-formalization” — they treated straightforward, plain-English resolutions with the heavy, multi-clause gravity of a tax code amendment. This stylistic mismatch introduced unnecessary complexity, causing human raters to favor the simpler, direct outputs generated by the general-purpose baselines. These outliers demonstrate that domain-specific models are highly calibrated tools: they are profoundly effective at deciphering dense legislative mazes, but can introduce unnecessary friction when applied to trivial text.

10.3 Automated Evaluation Alignment and AI Judge Viability

To determine whether advanced Large Language Models can serve as scalable proxies for expensive human legal evaluators, an automated ”LLM-as-a-Judge” framework was deployed to evaluate the identical dataset of machine-generated summaries. Using two independent judge families (Claude and Gemini) across 240 distinct evaluation runs, the AI judges scored the summaries on the same dimensions of Faithfulness, Clarity, and Usefulness. This section explores the quantitative alignment between human consensus and AI evaluation, examining both the systemic biases of the automated judges and their internal reliability.

10.3.1 Macro-Level Alignment and the ”Harshness” Bias

Comparing the AI-generated macro-averages against the human baseline reveals that automated judges generally reflect human preferences but apply a much stricter grading curve. Across all models, human evaluators tended to assign scores in the high 3.0 to 4.5 range, whereas the AI judges anchored their scores significantly lower, primarily in the 2.0 to 3.5 range. The table below showcases the average scores for each model across the AI evaluation framework.

Model Class	Average Faithfulness	Average Clarity	Average Usefulness
BART_base (Baseline)	2.00	3.33	2.08
LongT5_base	**3.12**	**3.35**	2.98
Legal_LED_base	3.10	3.23	**3.10**
Legal_LED_large	2.38	**3.35**	2.72

Despite this systemic ”harshness bias,” the AI framework successfully validated the core thesis identified by human reviewers: standard architectures fail at long-context legal summarization. According to the AI judges, the baseline BART_base model plummeted to a Faithfulness score of 2.00 and a Usefulness score of 2.08. By contrast, the long-context configurations excelled: LongT5_base achieved an Overall Mean of 3.15, practically tied with the domain-specific Legal_LED_base, which scored 3.14.

When analyzing document-by-document victories, the automated judges mirrored the human conclusion regarding architectural superiority. Out of 15 statutory documents evaluated, the AI identified Legal_LED_base as the overall winner in 7 instances, and LongT5_base in 6 instances. Legal_LED_large secured 2 victories, while the baseline BART_base failed to win a single document.

This algorithmic harshness bias also explains a notable ranking divergence between the human and AI evaluations regarding the Legal_LED_large architecture. In the human evaluations (Section 10.2), Legal_LED_large was highly regarded, securing strong averages for Usefulness (3.55) and Faithfulness (3.35) while achieving near-perfect scores on exceptionally long documents. However, under automated evaluation, it dropped to third place overall.

This discrepancy exposes how automated judges penalize generative abstraction. Because Legal_LED_large possesses a significantly larger parameter space, it tends to synthesize and rephrase complex statutory mandates into more fluent, generalized prose. Human legal professionals intuitively recognized and appreciated this advanced linguistic coordination. The AI judges, however, acted as rigid, pedantic auditors. Whenever Legal_LED_large abstracted a concept rather than extracting the exact, hyper-specific clauses from the source text, the AI judges severely penalized the text, driving its Faithfulness score down to 2.38. Consequently, the AI’s rigidity in enforcing exact structural replication punished the model’s fluency, causing it to underperform in the automated rankings despite its strong reception by human experts.

10.3.2 Correlation Analysis: Where AI Succeeds and Fails

A direct correlation analysis between the aggregated human scores and the AI judge scores reveals exactly where LLM evaluation is trustworthy — and where it deviates from human legal perception.

Evaluation Metric	Pearson Correlation (ρ)	Alignment Strength
Faithfulness	0.770	Strong positive alignment
Usefulness	0.521	Moderate positive alignment
Clarity	-0.102	No correlation (disconnected)

- High Alignment on Faithfulness: The AI judge proved to be an excellent proxy for measuring factual accuracy. When a model omitted a crucial statutory clause, the AI judge was just as likely as a human law student to detect the error and penalize the summary.
- Moderate Alignment on Usefulness: While the AI correctly identified the worst-performing baseline summaries, it struggled to perfectly replicate how humans intuitively weigh the practical value of a summary's overall structure.
- The Clarity Disconnect: The starkest deviation occurred within the Clarity metric. Human experts perceived wide variance in readability — rewarding BART_base (4.34) and harshly penalizing LongT5_base (3.69). The AI judges, however, viewed the Clarity of all four models as nearly identical (scoring them all tightly between 3.23 and 3.35). This indicates that LLMs evaluate text "fluency" fundamentally differently than human readers, likely overlooking the dense syntactic fatigue that frustrates human legal professionals.

10.3.3 Internal Reliability and Inter-Judge Consensus

For an automated proxy to be viable, it must be internally consistent. The run data demonstrates that modern LLM judges possess remarkable internal stability. When tracking run-to-run consistency (evaluating the exact same prompt and text on a separate iteration), both judge families achieved exceptional correlation across all metrics.

Evaluation Metric	Claude (Run 1 vs Run 2)	Gemini (Run 1 vs Run 2)
Faithfulness	0.934	0.946
Usefulness	0.866	0.908
Clarity	0.918	0.877

Inter-rater reliability — the agreement between the two different AI judge models — was also highly stable. When comparing Claude's grading logic directly against Gemini's, the correlation for the overall mean score rested at a strong Pearson r of 0.735. Furthermore, the two distinct AI models agreed on the exact same "winning" summary for a given document 60% of the time (Exact Winner Agreement Rate), resulting in a moderate Cohen's Kappa of 0.47 on winner categorization.

10.3.4 The Cost-Benefit Calibration

Ultimately, the data from this 240-run evaluation framework confirms that an LLM-as-a-Judge protocol is a highly effective, low-cost proxy for rapid benchmarking in the legal domain — with one specific caveat. Because the automated judges demonstrated a 0.770 correlation with human experts on Faithfulness, researchers can confidently deploy AI judges to mathematically verify whether a model is hallucinating or dropping statutory facts. The exceptionally high intra-judge stability (Pearson $r > 0.93$) guarantees that these automated evaluations are repeatable and robust. However, due to the total lack of correlation regarding text Clarity, final stylistic and syntactic polish must still be assessed by human legal professionals.

11 Discussion

The quantitative and qualitative findings presented in Chapter 10 uncover a tension between superficial text readability and substantive legal fidelity. While the metrics confirm that long-context, domain-specific architectures significantly outperform baseline models on aggregate metrics, a deeper examination of individual model failures reveals the precise structural vulnerabilities that complicate automated legal summarization. This chapter breaks down why these systems fail, what those failures teach us about evaluating legal language processing, and how future architectures must adapt to handle the hyper-rigorous demands of statutory text.

11.1 The Mechanics of Failure

To understand the operational limits of these models, we must first analyze the literal structural failures that occurred during evaluation. The empirical data highlights a stark drop-off in performance for baseline architectures, driven by a combination of strict hardware limitations and the uniquely dense architecture of legislative drafting.

The Hardware/Architectural Wall
The most catastrophic failures observed in this study were not errors in stylistic choice or sentence construction, but absolute failures of data ingestion. Standard transformer architectures like BART_base operate with a rigid maximum context window of 1,024 tokens. When tasked with processing complex federal legislation — such as the Foreign Bank Enforcement Act or the Protection from Sexual Predators Act — these models run into a hard mathematical wall.

Because a typical congressional bill easily spans tens of thousands of words, a 1,024-token window captures only a fraction of the total document. The baseline model is forced to truncate the vast majority of the text before any neural processing even begins. This structural reality directly accounts for BART_base's dismal macro-averaged Faithfulness score of 2.00 and Usefulness score of 2.08. The model is essentially blind to 90% or more of the statutory framework it is being asked to summarize.

Concrete Patterns of Omission
When a model is throttled by context truncation, its failures manifest in two distinct, highly problematic patterns: the Truncation Blindspot and Extinguished Exceptions.

1. The Truncation Blindspot
Legislative drafting follows a highly structured hierarchy. Preamble, intent, and high-level definitions sit at the very beginning of a bill, while operational mechanics, enforcement mechanisms, funding appropriations, and implementation deadlines are pushed to the back. When a short-context model truncates a bill like the Trafficking Prevention in Foreign Affairs Contracting Act, it captures the noble intent statement at the opening but completely drops the back-end compliance penalties. The resulting summary reports what the bill wants to achieve but entirely omits how the bill enforces it, rendering the summary functionally useless to a legal professional trying to assess compliance risk.

2. Extinguished Exceptions
Even more perilous than missing entire structural blocks is the systematic erasure of conditional clauses and legal exceptions. In statutory writing, an absolute mandate is frequently modified by subsequent operational exceptions. A model that lacks a complete context window or fails to resolve long-range semantic dependencies will often capture the baseline prohibition while dropping the vital qualifying language.

Example of a Legal Extinguishment Failure:

- Source Text Context: ”It shall be unlawful for any banking institution to engage in transaction X... [thousands of tokens later]... Provided, however, that the provisions of this section shall not apply to institutions operating under an approved micro-finance charter during their first five fiscal years.”
- Truncated Model Summary: ”The Act establishes a strict prohibition on banking institutions engaging in transaction X.”

In this scenario, the summary is syntactically flawless and perfectly readable, but legally catastrophic. By extinguishing the trailing exception, the model converts a conditional framework into a blanket ban. For a compliance officer or a practicing attorney, relying on such a summary introduces severe legal risk, as it active misrepresents the scope and bounds of the law. The failure of these models is not a failure of English grammar; it is a structural failure to preserve the complete logical architecture of the statute.

11.2 The Simplicity Premium: Why Truncation Drives Surface Readability

A striking finding from the evaluation metrics is the performance of the baseline architecture, BART_base, on the clarity metric. Despite failing catastrophically to preserve the factual integrity and completeness of the statutes — as evidenced by its low Faithfulness score — BART_base achieved an exceptionally competitive macro-averaged Clarity score of 3.33. Rather than a paradox, this high readability score is a predictable byproduct of severe context truncation.

Explaining the Clarity Metric
The clarity metric in the evaluation framework was explicitly designed to measure a single, isolated attribute: how easy the summary text is to read and comprehend on a surface linguistic level. It does not penalize factual omission or legal misstatement. When viewed through this specific lens, the baseline model’s high score is not an anomaly, but a reflection of structural simplicity. Because BART_base operates under a restrictive 1,024-token context window, it is fundamentally incapable of processing the dense operational machinery, complex procedural rules, or technical exceptions that populate the latter sections of legislative bills.

Consequently, the model builds its summaries almost exclusively from the introductory sections of the bills. In federal drafting, these opening sections typically consist of high-level preambles, broad purpose statements, and introductory titles (e.g., stating the general intent to expand a program or crack down on a specific offense). This introductory language is naturally much closer to standard English syntax and contains far fewer nested logical clauses

than the operational statutes that follow it. By summarizing only the simple text, BART_base naturally yields an innately simpler summary.

The Strucural Trade-off
This phenomenon exposes a critical structural trade-off in domain-specific natural language processing: the ”Simplicity Premium.” By pruning away 90% or more of a document’s legal complexity through context truncation, a short-context model produces a short, clean, high-level overview. The text flows naturally, avoids dense legal jargon, and is easy to digest — fully satisfying the criteria for surface-level clarity.

However, this readability is bought at the direct expense of legal utility. In a professional legal environment, a summary that is highly readable but missing the substantive core of the statute is deeply counterproductive. The data shows that while long-context architectures like LongT5_base and Legal_LED_base had to wrestle with the full structural noise and complex syntax of entire statutory documents — which naturally introduces more structural density into their outputs — they managed to maintain strong clarity scores (3.35 and 3.23 respectively) while providing significantly higher factual fidelity. This demonstrates that while short-context models achieve superficial clarity by evading the text’s difficulty, true architectural competence belongs to models that process the full legislative scope without sacrificing comprehensibility.

11.3 The Abstraction Penalty: Why AI Judges Disagreed with Human Legal Experts

The statistical divergence between human evaluation and the automated LLM-as-a-Judge framework reveals a fundamental methodological conflict in how human experts and artificial intelligence evaluate legal summaries. The most striking manifestation of this conflict is the ”Abstraction Penalty” suffered by the Legal_LED_large model. In the human trials, legal professionals and law students highly valued the large-scale architecture, awarding it strong marks for Usefulness (3.55) and systemic Clarity (4.31). Yet, within the automated evaluation framework, Legal_LED_large dropped to a distant third place overall, finishing with an inferior macro-averaged Overall Mean of 2.82 and a severely suppressed Faithfulness score of 2.38. Deconstructing this disagreement exposes the boundaries of current LLM-based evaluation protocols.

Rigid Auditing vs. Semantic Abstraction
The roots of this evaluation gap lie in the architectural differences between base-sized and large-sized models. With its expanded parameter space, Legal_LED_large transitions from basic extractive summarization — where a model simply isolates and copy-pastes key sentences from the source text — to true abstractive synthesis. The model actively reorganizes, rephrases, and consolidates dense, repetitive statutory language into sophisticated, macro-level prose.

When human legal experts read these abstractive summaries, they bring deep domain knowledge to the page. They possess the cognitive framework to recognize that if a summary rephrases a winding, multi-clause statutory sentence using elegant, higher-level legal terminology, the summary has successfully preserved the core legal intent. Humans reward

this cognitive compression because it reduces their mental workload, directly boosting their perception of clarity and practical utility.

Automated LLM judges, however, do not evaluate text with intuitive domain expertise; instead, they operate as rigid, pedantic text-matching auditors. When prompted to assess ”faithfulness” and ”accuracy,” the automated judges evaluate the text by scanning for direct semantic overlap, explicit structural signposts, and the exact statutory nomenclature of the original bill.

If Legal_LED_large synthesizes a complex series of clauses using an elegant synonym or groups multiple conditional rules under a unified conceptual umbrella, the AI judge fails to recognize the validity of the synthesis. Because the rephrased text lacks the exact, hyper-specific legal nouns and verbatim structures found in the source text, the automated judge flags these abstractive passages as unsupported claims or factual omissions. The AI judge effectively imposes a severe penalty on abstractive synthesis, mistaking sophisticated rephrasing for structural unfaithfulness.

The Extractive Bias of Automated Judges

This abstraction penalty highlights a systematic ”extractive bias” inherent to the LLM-as-a-Judge protocol. Models like Legal_LED_base and LongT5_base generated summaries that were structurally much closer to the original, fragmented language of the bills. Because they relied heavily on extracting chunks of verbatim text rather than synthesizing it, they easily satisfied the AI judge’s rigid verification criteria, achieving much higher automated Faithfulness scores of 3.10 and 3.12 respectively.

This creates an alarming methodological paradox for legal NLP research:

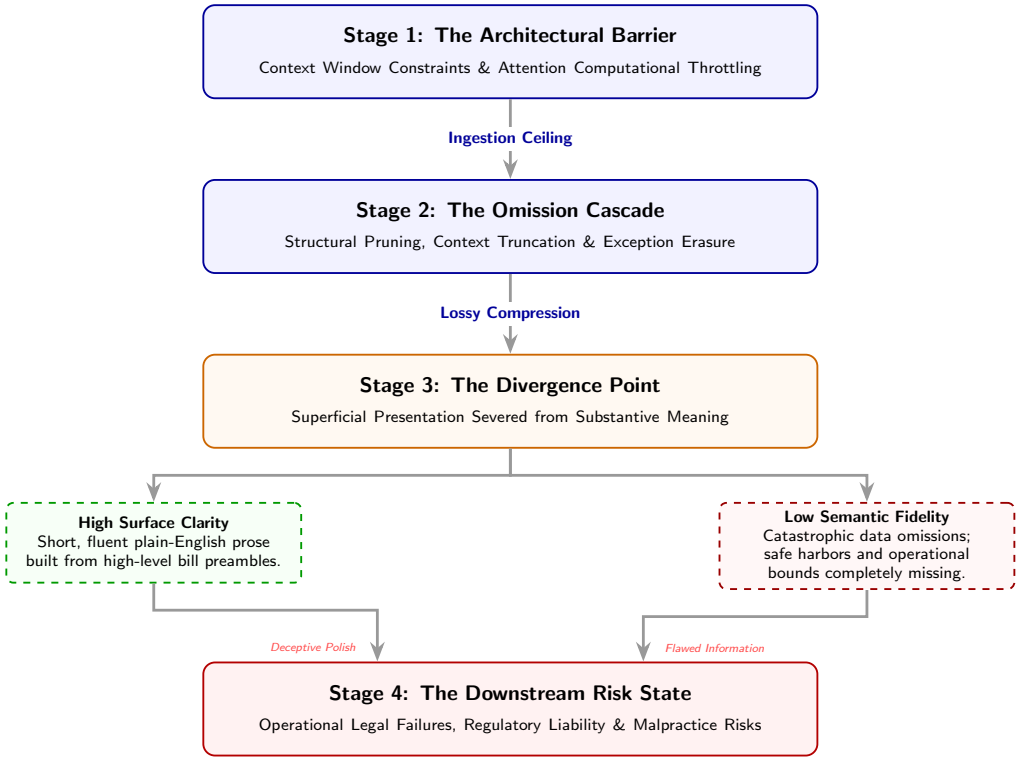
- An NLP model that copies and pastes fragmented, difficult-to-read statutory chunks will be rewarded by an AI judge for being highly ”faithful.”
- An NLP model that performs advanced, human-like abstractive synthesis will be heavily penalized by the same AI judge for being ”unfaithful.”

If automated metrics actively punish the exact linguistic behaviors that human legal practitioners prefer, relying exclusively on LLM judges will inadvertently bias model development backward — incentivizing the production of rigid, extractive models over fluid, truly intelligent abstractive legal assistants. While the automated judge remains a powerful tool for catching literal hallucinations or total context dropouts, the Legal_LED_large discrepancy proves that it cannot yet serve as a standalone replacement for human legal reasoning when evaluating advanced text synthesis.

11.4 A Conceptual Map of the Legal Summarization Failure Pipeline

To synthesize how isolated technical limitations compound into profound operational vulnerabilities, the findings of this study can be organized into a unified conceptual framework: The Legal Summarization Failure Pipeline. This linear pipeline illustrates how a low-level hardware or architectural constraint cascades downward through a sequence of algorithmic behaviors,

ultimately translating into real-world legal risk for the end user. By mapping out this progression, we can isolate the exact friction points where machine learning models diverge from the rigorous requirements of legal doctrine.



Stage 1: The Architectural Barrier (Input Context Throttling)
The pipeline originates at the hardware and algorithmic level. In standard transformer architectures, the fundamental bottleneck is the context window length and the computational complexity of the attention mechanism. If a model is restricted to a tight 1,024-token window or utilizes a dense attention mechanism that cannot scale quadratically to accommodate lengthy inputs, it encounters a hard physical boundary. This initial stage is completely independent of the text's grammar or content; it is a raw capacity ceiling that limits the volume of data the neural network can ingest.

Stage 2: The Omission Cascade (Structural and Conditional Pruning)
Once the model encounters the architectural barrier, it is forced to execute lossy compression routines, triggering the omission cascade. This manifests in two ways depending on the model's structure. Short-context architectures simply execute hard truncation, discarding all text past the token limit and creating a blindspot for any statutory text located in the latter sections of the document. Conversely, underperforming long-context models may suffer from

attention dilution, where the model registers the text but fails to resolve long-range semantic dependencies.

In both instances, the algorithmic result is identical: the model purges the complex logical framework of the statute. It prunes away trailing operational clauses, implementation deadlines, enforcement penalties, and crucial conditional exemptions — retaining only the structurally linear, high-level statements located near the beginning of the file.

Stage 3: The Divergence Point (Superficial Readability vs. Semantic Fidelity)
The third stage represents a profound deception in natural language generation. Because the model has stripped away the dense, multi-clause structural noise of the legal text, the resulting text profile shifts dramatically. The model outputs a summary that is short, syntactically simple, and composed of straightforward, plain-English sentences.

This creates a severe divergence point during evaluation:

- The Clarity Vector: Because the output is simple and uncluttered by exceptions, human readers and basic readability metrics award the model exceptionally high marks for surface-level clarity and fluency.
- The Fidelity Vector: Because the output has omitted the core statutory boundaries, exceptions, and conditions, the actual legal fidelity of the document drops to near zero.

The summary appears highly polished and authoritative precisely because it has ignored the difficult, necessary parameters of the underlying law.

Stage 4: The Downstream Risk State (Operational Legal Failure)
The final stage of the pipeline occurs when this deceptively fluent output is introduced into a live, professional legal workflow. A practicing attorney, compliance officer, or legal researcher reads the summary and, noting its exceptional surface clarity, misinterprets it as a comprehensive, accurate representation of the statutory mandate.

Because the summary has extinguished the legal exceptions (e.g., reporting a blanket prohibition while omitting the specific safe harbors or conditional use cases), the professional acts on incomplete information. This introduces massive downstream operational risks, ranging from non-compliance and regulatory fines to structural malpractice. The pipeline illustrates a sobering reality: in domain-specific natural language processing, a technical limitation at Stage 1 does not merely degrade text quality — it directly constructs a real-world liability at Stage 4.

11.5 Charting the Interventions: A Prelude to Future Work

The systematic vulnerabilities mapped out in the Legal Summarization Failure Pipeline (Section 11.4) make one thing clear: resolving the tension between surface clarity and legal fidelity requires more than simply expanding parameter counts or stretching context windows. While long-context models like Legal_LED and LongT5 overcome the hard truncation barriers of standard baselines, current architectures still struggle with attention dilution and rigid, block-arbitrary text processing.

Overcoming these remaining friction points demands fundamental shifts in how domain-specific models ingest statutory text and how automated systems evaluate abstractive outputs. While the exact technical execution and experimental parameters are detailed comprehensively in the upcoming Future Work chapter, three immediate, high-level interventions emerge directly from our discussion:

- Structure-Aware & Syntax-Guided Ingestion: Moving beyond arbitrary token-block attention (even in 16k-window models) toward dynamic, legal-hierarchical attention mechanisms that chunk, route, and prioritize text based on actual statutory boundaries (sections, sub-clauses, and stipulations).
- De-Biased Evaluation Frameworks: Designing next-generation LLM-as-a-Judge protocols equipped with advanced semantic alignment capabilities, effectively eliminating the ”abstraction penalty” by accurately distinguishing between elegant legal synthesis and true factual omission.
- Hybrid Neuro-Symbolic Generators: Integrating deterministic, rule-based legal parsing loops into neural architectures to ensure that conditional exceptions, safe harbors, and cross-references are tracked with perfect mathematical precision.

By addressing these core friction points, the legal NLP domain can transition away from deceptive surface readability and move toward genuine, high-fidelity statutory synthesis.

12 Conclusion

This thesis investigated the architectural and evaluation challenges of automated legal document summarization, specifically addressing the failure of standard short-context models on dense legislative bills. By scaling input context windows up to 16,384 tokens and introducing a three-tiered evaluation framework, this work evaluated the interplay between context length, domain adaptation, and evaluation metric reliability.

Research Question 1: The Impact of Input and Output Token Length Caps
To overcome the severe ”truncation penalty” inherent to standard 1,024-token transformer architectures, this study expanded input context limits using sparse-attention models (LED and LongT5). Furthermore, generation length caps were evaluated at 128 and 256 tokens. The empirical results demonstrate that while 128-token summaries offered high conciseness and readability, expanding the generation cap to 256 tokens delivered superior legal fidelity. Crucially, 256-token summaries maintained strong human readability while achieving consistently higher performance across ROUGE, BERTScore, and BLANC metrics, proving that additional decoding capacity is required to retain complex statutory conditions without overwhelming human readers.

Research Question 2: Domain Adaptation vs. Architectural Scaling
How do domain-adapted sparse-attention models (e.g., Legal-LED) compare against general-purpose long-context architectures (LongT5) and short-context baselines (BART)?

Domain-adapted pre-training proved to be just as critical as context window length. Models combining legal vocabulary pre-training with sparse attention systematically outperformed both short-context baselines (BART) and non-specialized long-context architectures (LongT5). Across the experimental benchmark, **LED-base emerged as the most consistent overall model**, delivering balanced, top-tier performance across lexical, semantic, and qualitative evaluation dimensions without exhibiting destabilizing trade-offs.

Research Question 3: Metric Reliability and Multi-Tiered Evaluation
Can automated metrics adequately capture legal summary quality, or are multi-tiered evaluation methods—including human experts and AI evaluators—necessary?

Standard surface-level lexical overlap metrics (ROUGE) mask critical quality differences in legal text, making a three-tiered framework (lexical, reference-free semantic via BLANC, and human/AI qualitative judges) necessary. This multi-tiered setup uncovered a distinct abstraction penalty: while LED-large scored highest on automated metrics and earned top ratings from human legal experts, it suffered a performance drop under the AI-as-a-Judge framework. Because LED-large produced more fluid, abstractive syntheses rather than verbatim extractions, the AI judge penalized its fidelity—a penalty that human legal experts did not replicate. This divergence reveals that while AI evaluators serve well as rapid screening assistants, they remain unreliable as standalone legal judges due to their tendency to misinterpret valid abstractive rephrasing as factual deviation.

Key Findings and Insights
The core empirical insights gained from this study include:

- The 16,384 / 256 Parameter Baseline: Coupling a 16,384-token sparse input context with a 256-token generation cap provides the ideal structural balance for preventing information loss in legislative bills while maintaining human readability.
- The Abstraction Penalty Dynamics: Higher abstractive capability in larger models (LED-large) can artificially degrade scores in LLM-driven evaluation frameworks despite earning human legal approval, highlighting a misalignment between AI evaluators and human expert judgement.
- Deployment Viability: Translating these empirical results into the interactive LexiBrief web tool demonstrates that sparse-attention long-context models are operationally ready to assist policy analysts and compliance teams in real-world workflows.

13 Future Work

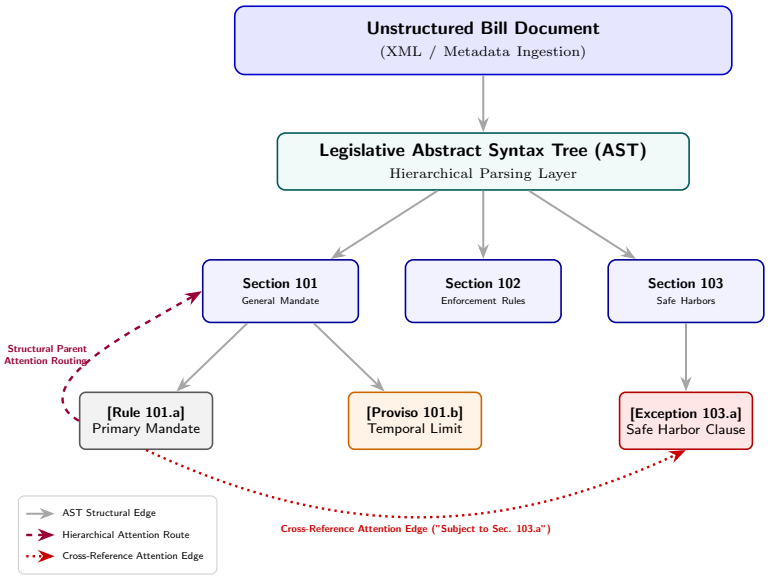
The empirical findings and failure modes identified in Chapter 10, combined with the structural analysis of the Legal Summarization Failure Pipeline in Chapter 11, demonstrate that scaling parameter counts alone cannot resolve the fundamental challenges of statutory summarization. Building upon the high-level interventions introduced in Section 11.5, this chapter dives deeper into the specific architectural, methodological, and experimental frameworks required to advance state-of-the-art legal NLP.

Sections 13.1 through 13.3 systematically expand upon the three technical proposals first introduced in the Discussion chapter: structure-aware attention mechanisms, de-biased evaluation protocols, and neuro-symbolic generation constraints. In addition to these core pillars, Section 13.4 introduces a fourth, vital research avenue not previously detailed in Chapter 11 — the development of specialized, exception-dense legal benchmarks specifically designed to stress-test future summarization models.

13.1 Structure-Aware & Syntax-Guided Ingestion

While long-context transformer architectures evaluated in this study successfully bypass the catastrophic context truncation seen in standard baselines, their attention mechanisms remain fundamentally blind to the formal structure of legislative drafting. Existing long-context architectures divide input sequences into arbitrary, fixed-size token blocks (e.g., 512 or 1,024-token sliding windows or sparse global blocks). In statutory writing, however, semantic boundaries do not align with arbitrary token counts; they align with structural hierarchy—chapters, sections, sub-clauses, and provision statements.

Future legal architectures must replace arbitrary token-block attention with Structure-Aware & Syntax-Guided Ingestion. Rather than treating a bill as a continuous sequence of unformatted tokens, the model's encoder should operate over an Abstract Syntax Tree (AST) generated from the document's formal legislative XML or metadata schema.



Technical Mechanism: Hierarchy-Guided Sparse Attention
Instead of calculating uniform global or local sliding attention across all tokens, a syntax-guided encoder enforces structural attention masks based on the document's syntactic tree:

- Node-Level Local Attention: Tokens within a specific sub-clause (e.g., a specific statutory restriction or definition) compute dense local self-attention to capture exact clause-level mechanics.
- Structural Parent Routing: Sub-clause nodes route summary tokens directly up to their parent section and chapter nodes, ensuring that qualifying conditions and exceptions remain explicitly bound to the primary legal mandate they modify, regardless of token distance.
- Cross-Reference Attention Edges: When a statute contains explicit cross-references (e.g., "subject to the provisions of Section 4(b)(2)"), dynamic attention edges are established directly between the referencing clause and the target statutory node.

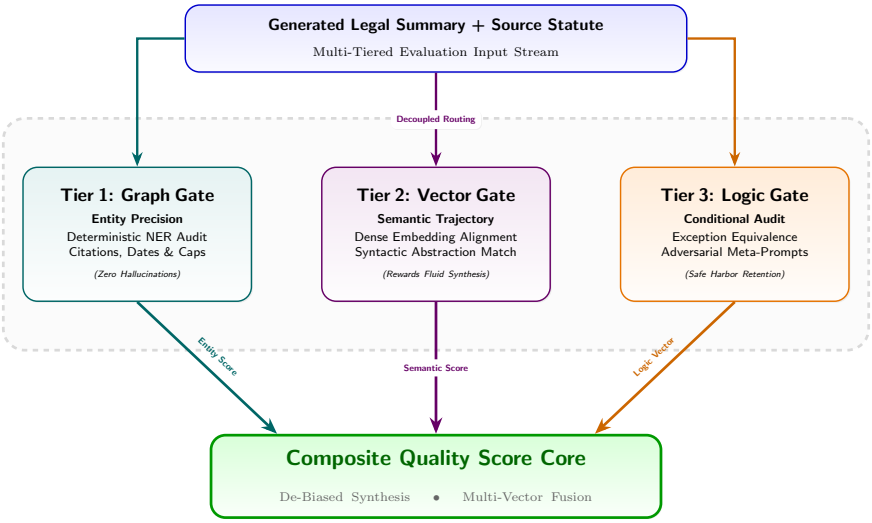
By embedding the document's logical hierarchy directly into the attention matrix, the transformer no longer wastes attention bandwidth across non-relevant text blocks. This structural grounding prevents the attention dilution that currently causes models to overlook deeply nested exception clauses during long-context processing.

13.2 De-Biased Evaluation Frameworks

The statistical divergence documented in Section 11.3 exposed a fundamental flaw in contemporary legal NLP evaluation: single-prompt automated LLM judges exhibit a pronounced "extractive bias." When evaluating complex summaries, automated judges routinely punish

high-performing, abstractive models for synthesizing complex legalese into elegant prose, while rewarding lower-quality extractive models that simply copy and paste fragmented source text.

To bridge this gap without relying entirely on cost-prohibitive, time-intensive human legal audits, future research must replace monolithic LLM evaluation prompts with Multi-Tiered, De-Biased AI Evaluation Frameworks. These architectures decouple the evaluation of hard factual precision from the evaluation of abstractive semantic synthesis.



The figure above illustrates the three-tiered evaluation framework.

Tier 1: Deterministic Knowledge Graph Auditing (Factual & Entity Precision)
The first tier operates deterministically to eliminate the hallucination blindspots of neural evaluation. Before passing text to an LLM, a domain-specific Named Entity Recognition (NER) pipeline extracts all rigid legal entities from both the source statute and the generated summary:

- Statutory citations (e.g., 18 U.S.C. §1030)
- Enforcement deadlines and effective dates
- Monetary thresholds, penalty amounts, and statutory caps
- Defined legal entities and party designations

A knowledge graph alignment algorithm verifies whether every entity in the summary directly maps to a corresponding node in the source document. If a model modifies a critical statutory figure or omits a mandatory legal term, Tier 1 flags the discrepancy immediately. By

handling factual verification deterministically, this tier relieves the LLM judge from performing string-matching arithmetic — a task LLMs execute unreliably.

Tier 2: Semantic Trajectory & Vector Alignment (Abstractive Synthesis Quality)
To evaluate fluid, high-level legal synthesis without triggering the abstraction penalty, Tier 2 replaces surface token matching with dense vector trajectory alignment.

Using specialized legal embedding spaces (e.g., Legal-BERT or fine-tuned statutory sentence transformers), the framework measures the semantic distance between synthesized summary passages and their corresponding source sections. Because dense vector representations capture high-level conceptual meaning rather than verbatim word choices, a summary that rephrases complex statutory jargon using elegant, higher-level legal terminology scores exceptionally high in Tier 2. This explicitly incentivizes and rewards abstractive compression.

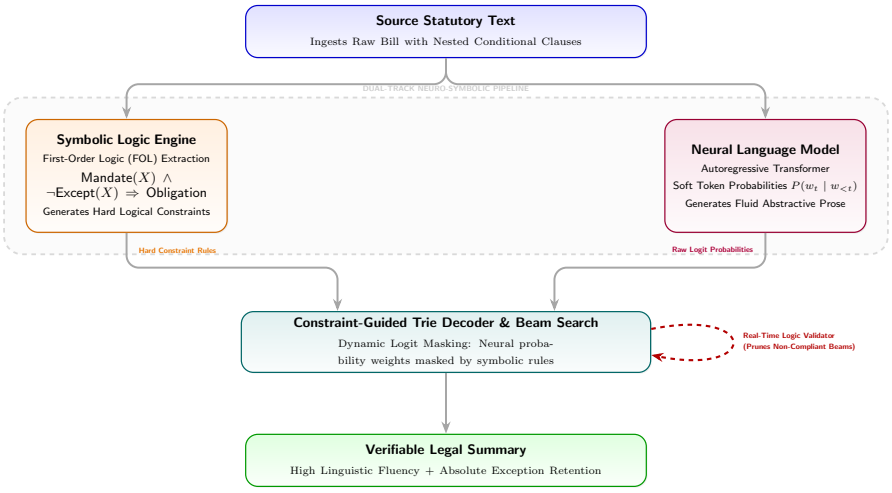
Tier 3: Constraint-Aware Meta-Evaluation (Logic & Exception Auditing)
The final tier addresses the critical ”Extinguished Exceptions” problem identified in Section 11.1. Instead of asking a single prompt to judge ”Overall Faithfulness,” Tier 3 deploys targeted, adversarial meta-prompts designed solely to audit logical equivalence.

Tier 3 Audit Prompt Structure
”Identify all conditional clauses (’unless’, ’except’, ’provided that’) in Source Clause A. Does Summary Passage B retain the exact logical boundaries of these conditions? Answer yes/no and output a formal logical truth table.”

By evaluating text through three distinct, specialized channels, this multi-tiered paradigm eliminates extractive bias. It ensures that next-generation legal NLP models are evaluated by AI judges that value sophisticated, human-like legal synthesis while maintaining zero tolerance for factual misrepresentation or exception erasure.

13.3 Hybrid Neuro-Symbolic Generators

Standard transformer architectures generate text autoregressively by sampling from a soft probability distribution over tokens. While this purely statistical mechanism excels at generating fluent, natural language, it possesses no native concept of boolean logic or legal necessity. In statutory summarization, a single omitted token — such as converting ”shall not” to ”shall” or dropping an ”unless” clause — flips the legal meaning of a statute entirely. To guarantee that generated summaries adhere strictly to the underlying statutory logic, future research must shift from unconstrained neural generation toward Hybrid Neuro-Symbolic & Constraint-Guided Generation.This paradigm combines the fluid linguistic capabilities of deep neural networks with the non-negotiable, deterministic precision of symbolic logic engines.



First-Order Logic (FOL) Rule Extraction
Before text generation begins, a symbolic parsing engine analyzes the source bill and converts its conditional framework into formal logical predicates. Statutory provisions are mapped into boolean conditional trees:

$$\text{Mandate}(X) \land \neg\text{Disqualified}(X) \implies \text{Obligation}$$

These extracted logical rules define the non-negotiable ”hard constraints” of the statute — the mandatory preconditions, statutory exceptions, and safe harbor parameters that must be reflected in any faithful summary.

Lexically Constrained Trie Decoding
During the generation phase, the model’s soft decoding process (e.g., beam search or top-p sampling) is dynamically governed by a symbolic constraint mask. If the neural model attempts to generate a summary passage covering a specific statutory rule, the decoding trie enforces hard lexical and logical bounds:

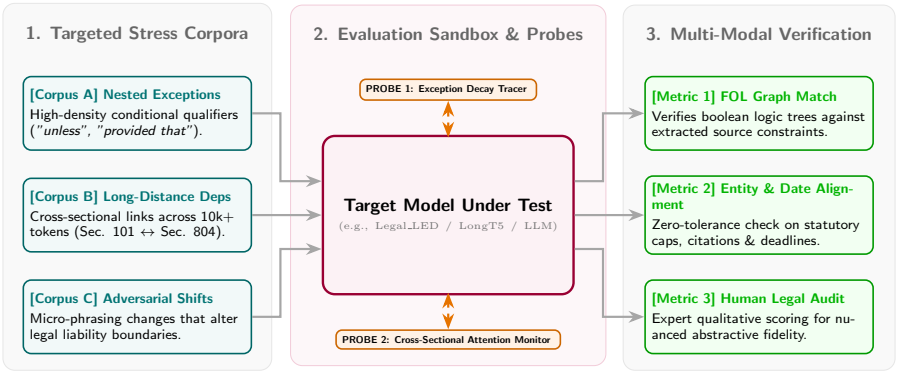
- Mandatory Anchor Injection: If a statutory exception is triggered in the source text, the decoder forces the probability distribution of tokens to prioritize phrases that preserve the conditional boundary (e.g., ”provided that,” ”except where,” ”subject to Section B”).
- Logical Masking: Tokens that would convert a conditional obligation into an absolute prohibition (e.g., dropping qualifying clauses) receive a probability weight of zero, effectively blocking unfaithful phrasing before it can be generated.

Verification-Guided Correction Loops
Rather than relying on post-hoc error correction, neuro-symbolic generators incorporate a real-time logical validator directly into the inference loop. As candidate summary sentences are constructed in beam search, they are evaluated against the extracted First-Order Logic constraints. If a candidate sentence violates a core statutory rule, the beam is instantly pruned and rerouted toward a logically sound alternative.

By anchoring statistical natural language generation inside a deterministic symbolic framework, hybrid neuro-symbolic systems eliminate the risk of "extinguished exceptions," producing summaries that are both linguistically natural and legally infallible.

Specialized Legal Benchmarks & Stress-Testing Corpora
Beyond advancing model architectures and automated judges, the legal NLP community faces a foundational evaluation bottleneck: existing public datasets are fundamentally unequipped to measure legal fidelity. Standard legislative summarization benchmarks, such as BillSum or GovReport, rely primarily on high-level executive summaries written by legislative staff. While useful for general NLP tasks, these reference summaries are often written to explain political intent rather than preserve precise legal mechanics. Consequently, models evaluated on these datasets are rewarded for high-level topical overlap and surface-level fluency, while catastrophic legal failures — such as the complete omission of a safe harbor clause — go entirely unpenalized by standard metrics like ROUGE or BERTScore.

To drive meaningful progress, future work must establish purpose-built, open-access Legal Stress-Testing Corpora. Rather than evaluating models on random collections of legislative bills, these specialized benchmarks must be curated around high-risk structural edge cases designed to deliberately challenge neural attention mechanisms. Below is a diagram of a proposed framework.



High-Density Nested Exception Datasets
Standard datasets contain a mix of routine procedural bills and complex statutes. A stress-testing benchmark must isolate bills containing an exceptionally high density of conditional qualifiers ("unless," "provided that," "except as stipulated in," "subject to"). By evaluating

models specifically on high-density exception corpora, researchers can measure a model's Exception Decay Rate (the exact frequency with which a model prunes away critical legal boundaries as input length increases).

Long-Distance Dependency Corpora

To evaluate whether long-context transformers actually resolve structural relationships across thousands of tokens, benchmark corpora must include bills where a primary rule established in an early section (e.g., Section 101) is fundamentally altered or negated by a provision located near the end of the document (e.g., Section 804). Ground-truth annotations for these corpora must map explicit links between referencing and referenced clauses, enabling fine-grained measurement of cross-sectional attention retention.

Logic-Tree Ground Truths & Fine-Grained Annotations

Future benchmarks must move beyond providing a single, human-written reference text. Instead, benchmark samples should be paired with multi-modal ground truths as follows:

- Natural Language Reference Summaries: Human-written summaries annotated at the sentence level for factual vs. abstractive content.
- First-Order Logic (FOL) Graphs: Formal logical trees representing the exact statutory rules, obligations, and exceptions within the bill.
- Entity Knowledge Graphs: Extracted mappings of defined terms, statutory penalties, and jurisdictional boundaries.

By evaluating candidate summaries against formal logic trees rather than plain text alone, these specialized benchmarks will establish a rigorous, objective standard for statutory NLP—ensuring that next-generation models are measured not by how fluent they sound, but by how accurately they preserve the rule of law.

14 Appendix

14.0.1 Evaluation Results Table

Model	Config	FK	DC	R1	R2	RL	BERT	BLANC
BART base	512/128	34.40	14.21	0.42	0.24	0.32	0.82	0.14
Long T5 base	512/128	40.65	14.92	0.45	0.25	0.34	0.82	0.16
Legal LED base	512/128	39.68	15.03	0.43	0.24	0.33	0.81	0.15
Legal LED large	512/128	39.88	14.92	0.43	0.23	0.31	0.81	0.15
BART base	512/256	36.93	14.55	0.43	0.24	0.32	0.82	0.14
Long T5 base	512/256	67.01	18.26	0.43	0.22	0.32	0.80	0.16
Legal LED base	512/256	53.20	16.70	0.42	0.22	0.32	0.80	0.15
Legal LED large	512/256	67.45	18.44	0.46	0.22	0.30	0.81	0.17
BART base	1024/128	35.78	14.35	0.45	0.26	0.34	0.83	0.16
Long T5 base	1024/128	42.93	15.11	0.47	0.28	0.36	0.82	0.18
Legal LED base	1024/128	39.87	14.92	0.45	0.26	0.35	0.82	0.16
Legal LED large	1024/128	41.83	15.14	0.47	0.27	0.35	0.83	0.18
BART base	1024/256	42.44	15.22	0.46	0.26	0.34	0.83	0.17
Long T5 base	1024/256	68.17	18.31	0.49	0.29	0.37	0.82	0.20
Legal LED base	1024/256	61.57	17.66	0.46	0.26	0.35	0.82	0.18
Legal LED large	1024/256	68.13	18.55	0.48	0.25	0.33	0.82	0.19
Long T5 base	2048/128	42.07	15.02	0.50	0.31	0.39	0.83	0.19
Legal LED base	2048/128	42.19	15.13	0.49	0.31	0.38	0.83	0.19
Legal LED large	2048/128	41.06	15.09	0.48	0.29	0.37	0.83	0.19
Long T5 base	2048/256	68.34	18.36	0.53	0.33	0.40	0.83	0.23
Legal LED base	2048/256	67.27	18.34	0.51	0.31	0.38	0.83	0.21
Legal LED large	2048/256	66.20	18.27	0.52	0.31	0.37	0.84	0.22
Long T5 base	4096/128	42.20	15.03	0.49	0.31	0.39	0.83	0.19
Legal LED base	4096/128	39.75	14.86	0.49	0.31	0.39	0.84	0.19
Legal LED large	4096/128	41.84	15.15	0.51	0.32	0.39	0.84	0.20
Long T5 base	4096/256	66.34	18.13	0.53	0.34	0.41	0.84	0.23
Legal LED base	4096/256	65.39	18.13	0.51	0.31	0.39	0.83	0.21
Legal LED large	4096/256	72.18	19.01	0.55	0.33	0.39	0.84	0.24
Legal LED base	8192/128	41.67	15.11	0.50	0.32	0.39	0.84	0.19
Legal LED large	8192/128	43.44	15.29	0.50	0.31	0.38	0.84	0.20
Legal LED base	8192/256	68.12	18.41	0.53	0.33	0.40	0.84	0.23
Legal LED large	8192/256	68.45	18.49	0.55	0.36	0.40	0.84	0.25
Legal LED large	16384/128	43.13	15.39	0.51	0.32	0.39	0.84	0.20
Legal LED large	16384/256	61.87	17.80	0.56	0.35	0.42	0.85	0.24

14.0.2 Summary Notes

- **Configuration Format:** All models use 3 epochs, shown as input_length/output_length
- **Column Abbreviations:**
  - FK = Flesch-Kincaid Grade
  - DC = Dale-Chall Score
  - R1/R2/RL = ROUGE-1/2/L
  - BERT = BERTScore F1
  - BLANC = BLANC Help Score
- **Readability:** Higher FK and DC scores indicate more complex text
- **ROUGE/BERT/BLANC:** Range 0-1, higher is better

14.1 Survey Results Report

Below find the link to the pdf containing the full report on the results of the survey generated by Qualtrics.

https://online.fliphtml5.com/zwtkci/Qualtrics_report/

14.2 Full Exact Prompt Used for Synthetic LLM Judgement

```
You are a legal expert evaluator applying a fixed survey rubric to summaries of legal documents.
You are not a human survey participant. You are an automated judge completing the same structured evalua
Evaluate the summaries ONLY against the provided evidence excerpt.
Do not use outside knowledge.
Do not assume facts not present in the excerpt.
Do not reward a summary for sounding fluent if it is inaccurate.
The model identities are hidden. Treat Summary A, B, C, and D independently.

Rating scale:
1 = very poor
2 = poor
3 = acceptable
4 = good
5 = excellent

Criteria:
1. accuracy_faithfulness:
Does the summary accurately reflect the evidence and avoid incorrect or unsupported claims?
2. clarity:
Is the summary easy to read and understand?
3. usefulness:
```

```
Would this summary help someone quickly understand the key points?

DOCUMENT ID:
{doc_id}

EVIDENCE EXCERPT:
{excerpt}

SUMMARY A:
{summaries_by_label["A"]}

SUMMARY B:
{summaries_by_label["B"]}

SUMMARY C:
{summaries_by_label["C"]}

SUMMARY D:
{summaries_by_label["D"]}

Return JSON only using this exact structure:
{{
  "ratings": {{
    "A": {{
      "accuracy_faithfulness": 1,
      "clarity": 1,
      "usefulness": 1
    }},
    "B": {{
      "accuracy_faithfulness": 1,
      "clarity": 1,
      "usefulness": 1
    }},
    "C": {{
      "accuracy_faithfulness": 1,
      "clarity": 1,
      "usefulness": 1
    }},
    "D": {{
      "accuracy_faithfulness": 1,
      "clarity": 1,
      "usefulness": 1
    }}
  }}
}}
For the real evaluation, replace the example 1 values with appropriate scores from 1 to 5.
```

References

[1] https://fiscalnote.com/reports/2026-state-of-government-affairs-report/.

[2] https://www.analyticsvidhya.com/blog/2023/03/exploring-the-extractive-method-of-text-summarization/.

[3] https://www.ibm.com/think/tutorials/abstractive-text-summarization.

[4] Yang Liu and Mirella Lapata. "Text Summarization with Pretrained Encoders". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3730–3740. DOI: 10.18653/v1/D19-1387. URL: https://aclanthology.org/D19-1387/.

[5] Abigail See, Peter J. Liu, and Christopher D. Manning. "Get To The Point: Summarization with Pointer-Generator Networks". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Regina Barzilay and Min-Yen Kan. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1073–1083. DOI: 10.18653/v1/P17-1099. URL: https://aclanthology.org/P17-1099/.

[6] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. URL: https://aclanthology.org/2020.acl-main.703/.

[7] https://www.datacamp.com/blog/self-attention.

[8] Iz Beltagy, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer". In: *CoRR* abs/2004.05150 (2020). arXiv: 2004.05150. URL: https://arxiv.org/abs/2004.05150.

[9] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791. URL: https://ieeexplore.ieee.org/document/726791.

[10] Mandy Guo et al. "LongT5: Efficient Text-To-Text Transformer for Long Sequences". In: *Findings of the Association for Computational Linguistics: NAACL 2022*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 724–736. DOI: 10.18653/v1/2022.findings-naacl.55. URL: https://aclanthology.org/2022.findings-naacl.55/.

[11] https://news.mit.edu/2024/mit-study-explains-laws-incomprehensible-writing-style-0819.

[12] https://time.com/3858309/attention-spans-goldfish/.

[13] Anastassia Kornilova and Vladimir Eidelman. "BillSum: A Corpus for Automatic Summarization of US Legislation". In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Ed. by Lu Wang et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 48–56. DOI: 10.18653/v1/D19-5406. arXiv: 1910.00523 [cs.CL]. URL: https://aclanthology.org/D19-5406.

[14] Mohamed Elaraby and Diane Litman. "ArgLegalSumm: Improving Abstractive Summarization of Legal Documents with Argument Mining". In: *Proceedings of the 29th In-*

ternational Conference on Computational Linguistics. Ed. by Nicoletta Calzolari et al. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 6187–6194. URL: https://aclanthology.org/2022.coling-1.540/.

[15] Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. "Fill in the BLANC: Human-free quality estimation of document summaries". In: *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*. Ed. by Steffen Eger et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 11–20. DOI: 10.18653/v1/2020.eval4nlp-1.2. URL: https://aclanthology.org/2020.eval4nlp-1.2/.

[16] Lianmin Zheng et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. 2023. arXiv: 2306.05685 [cs.CL]. URL: https://arxiv.org/abs/2306.05685.

[17] https://huggingface.co/models.

[18] https://lightning.ai/.