An NLP Framework for Literature Summarization in Law and Policy

Long Paper (Final Report)

Atharva Kirkole, Yuxuan (Amber) Zhi, Mahnoor Sheikh, Vaibhav Vennam

Abstract

Accessing and interpreting legal and policy documents remains a significant challenge for individuals without legal expertise. Our project aims to develop an NLP framework for literature summarization in law and policy, focusing on empowering people who represent themselves in court or wish to understand regulatory frameworks. It also supports students, journalists, and citizens seeking to interpret complex legislation without relying solely on media perspectives.

The framework fine-tunes transformer-based summarization models on legislative datasets such as BillSum and GovReport, alongside a TF-IDF cosine similarity extractive baseline. The fine-tuned model demonstrates significant improvements over the baseline, showing increases of 55.6 percent in ROUGE-1, 91.7 percent in ROUGE-2, and 111.1 percent in ROUGE-L scores compared to the TF-IDF model.

A retrieval-only QA component enables users to ask targeted questions and receive the most relevant bill sections without generative modification, ensuring factual reliability and fast response times. The system is generated with a Gradio interface, enabling efficient summarization and navigation of legal texts even in resource-constrained environments.In addition, a dedicated Summarizer Interface allows users to upload or paste lengthy bills and receive concise, hierarchically processed summaries that preserve legal context and structure.

1 Introduction

Legal and policy documents are often lengthy, technical, and inaccessible to the general public. Lawyers and policymakers are trained to interpret such material, but individuals dealing with regulations or legal processes independently often lack the expertise to do so. This creates barriers to justice, learning, and transparency.

Automatic summarization of legislative and policy texts provides a means to reduce this gap by transforming complex content into simpler, human-readable summaries. While prior research has achieved strong results on domain-specific summarization, limited work has been done on interactive, user-oriented systems that democratize access to legal information.

Our project fills this gap by combining extractive and abstractive summarization models with a retrieval-based chatbot interface. Users enter queries about legal or policy issues (e.g., "tenant eviction laws" or "student debt relief") and receive concise summaries generated from relevant bills or reports.

2 Related Works

In previous research, Kornilova and Eidelman (2019) introduced the BillSum dataset, which includes U.S. Congressional and California state bills with human-written summaries.They identified that limited research applied automatic summarization to legislative text, even though it is widely used in other domains. They applied the Document Context Model (DOC) , the Summary Language Model (SUM), and an ensemble DOC+SUM. Their results showed that SUM provided the highest ROUGE scores for congressional bills. However, when applied to CA bill, it did not perform as well as on U.S. bills. Applying this dataset, Jain et al. (2021) conducted a comparative analysis of automatic summarization. They tested methods including Lexrank, Luhn, Edmundson, LSA, KL, SumBasic, TextRank, and Reduction, and evaluated them using ROUGE, BLEU, and cosine similarity. Their findings showed that graph-based approaches, such as LexRank and TextRank, consistently outperformed simple frequency-based methods. This emphasized the importance of modeling sentence level relationships in legislative summa-

rization. While the first two studies focused on extractive summarization, Shukla et al. (2022) used three datasets (IN-Abs, IN-Ext, UK-Abs) and applied both extractive and abstrative methods. Their study highlighted that DSDR and SummaRuNNer are strong extractive options, while Legal-Pegasus is good for abstractive summarization. For long documents, fine-tuning models through chunking showed promise, and they also pointed out that evaluating summaries in the legal domain requires input from experts, not just automatic metrics.

3 Dataset Description

3.1 Source

BillSum is a summarization of US Congressional and California state bills. The US bills were collected from the Govinfo service provided by the United States Government Publishing Office (GPO) under CC0-1.0 license. The California bills were collected from the 2015-2016 session from the legislature's website.

Government report dataset consists of reports written by government research agencies including Congressional Research Service and U.S. Government Accountability Office.

3.2 Features

The BillSum dataset consists of three parts: US training bills (18949), US test bills (3269) and California test bills (1237).

```
"text": bill text
"summary": human-written summary of the bill
"title": title of the bill
"text_len": number of chars in text
"sum_len": number of chars in summary
```

The GovReport dataset consists of 17517 training, 973 validation and 973 test observations.

```
"id": paper id
"report": body of the report
"summary": summary of the report
```

4 Methodology

Our methodology integrates legal document summarization with a conversational interface:

4.1 Data Collection & Preprocessing

- We used two primary datasets:
  - BillSum - a summarization dataset of U.S. Congressional and California state bills, containing bill texts and human-written summaries.
  - GovReport - a corpus of government research reports from the Congressional Research Service and U.S. Government Accountability Office, paired with summaries.
  - Both datasets were formatted to use consistent column names and structure so they could later be combined for model training.
- Text and Summary Cleaning
  - Each document and its summary were cleaned to remove boilerplate and formatting artifacts. Using regular expressions and BeautifulSoup, the preprocessing script removed page numbers, report headers, table of contents markers, rule lines, and HTML tags. Unicode normalization was applied to make punctuation and spacing consistent. The same process was applied to summaries, mainly to remove extra spaces and small formatting issues. The cleaned outputs were stored in new columns called cleaned text and cleaned summary for all dataset splits.
- Sentence Tokenization
  - The cleaned text was then split into sentences using NLTK function. Short headings were merged with the following sentence to keep context. The resulting sentences column supports extractive baselines and helps the model handle longer documents more effectively.
- Final Dataset Structure
  - After preprocessing, each dataset contained the following columns: text, summary, cleaned text, cleaned summary, and sentences.

4.2 Baseline Model: Extractive TF-IDF Cosine Similarity

To establish a reproducible baseline, we implemented an extractive summarization method based on TF-IDF weighting and cosine similarity, inspired by TextRank-style graph scoring. This baseline serves as a reference point for later fine-tuning transformer-based models.

This method scores each sentence based on its overall semantic similarity with all other sentences in the document, identifying those that best represent the main ideas. The most central and informative sentences are then selected and arranged in their original order to form a concise extractive summary. The approach was applied to the *GovReport* and *BillSum* combined test dataset (2607 observations) to generate the initial set of extractive summaries and evaluate the baseline performance.

The ROUGE-1, ROUGE-2, and ROUGE-L scores achieved are reported in Table 2 as shown in the Results section.

Although this provided a useful baseline, the ROUGE scores achieved by the fine-tuned model were exponentially higher, demonstrating dramatically improved content coverage and coherence compared to the TF-IDF approach.

4.3 Abstractive Modeling

We selected **DistilBART** (`sshleifer/distilbart-cnn-12-6`) as our primary abstractive summarization model due to its optimal balance between performance and computational efficiency. DistilBART is a distilled version of the BART (Bidirectional and Auto-Regressive Transformers) model, specifically designed for sequence-to-sequence tasks like summarization.

Key Advantages of DistilBART:

- **Parameter Efficiency**: With approximately 222 million parameters (60% reduction from original BART-large)
- **Training Speed**: Faster convergence and lower memory requirements
- **Strong Performance**: Maintains ∼97% of BART's performance on CNN/DailyMail summarization
- **Optimized Architecture**: 6 encoder layers and 6 decoder layers with reduced dimensionality

4.4 Legal Domain Adaptations

4.4.1 Specialized Decoding Strategies

We implemented decoding parameters specifically optimized for legal text:

- **Length Penalty (1.2)**: Encourages wordy comprehensive summaries
- **No Repeat N-gram (3)**: Prevents repetition common in legal documents
- **Beam Search (4-6 beams)**: Balances quality and computational cost
- **Early Stopping**: Prevents overly verbose legal jargon repetition

4.4.2 Handling Legal Document Structure

Our methodology specifically addresses challenges in legal text:

- Preservation of legal definitions and key terms
- Maintenance of conditional logic and provisions
- Extraction of actionable requirements and obligations
- Identification of temporal constraints and deadlines

4.5 Model Architecture Details

DistilBART employs a standard transformer encoder-decoder architecture with the following specifications:

Component	Specifications
Encoder/Decoder	6 layers, 768 hidden dimensions, 12 attention heads
Total Parameters	∼222 million
Vocabulary Size	50,265 tokens
Max. Length	1,024 tokens

Table 1: DistilBART Architecture Specifications

4.6 Fine-Tuning Strategy

4.6.1 Data Preparation and Preprocessing

We prepared our legal document summarization dataset by combining cleaned texts from both BillSum and GovReport datasets. The preprocessing pipeline involved several key steps:

First, we merged the training splits from both datasets to create a combined training set of legal documents. This approach allowed the model to learn from diverse legal writing styles and document structures. The datasets were formatted to

maintain consistent column names and structure across both sources.

For tokenization, we implemented a specialized preprocessing function that handled the unique characteristics of legal text. The input legal documents were tokenized with a maximum length of 512 tokens, while the target summaries were limited to 64 tokens. This balance allowed the model to process substantial legal content while generating concise summaries appropriate for the legal domain.

4.6.2 Training Configuration

We employed a comprehensive training strategy specifically tailored for legal text summarization. The training arguments were configured with evaluation and saving strategies set to occur every 100 steps, with logging every 50 steps to monitor progress closely. Due to computational constraints, we used a per-device batch size of 2 with gradient accumulation steps of 4, effectively creating a batch size of 8.

The model was trained for 3 epochs with a learning rate of 5×10^{-5} and weight decay of 0.01 for regularization. We enabled mixed precision training (fp16) to accelerate training while maintaining stability. Warm-up steps of 100 were included to gradually increase the learning rate at the beginning of training, preventing large gradient updates that could destabilize the fine-tuning process.

4.6.3 Tokenization and Input Formatting

We implemented careful tokenization to handle the unique characteristics of legal documents. The preprocessing pipeline was designed to accommodate the structural complexity and specialized vocabulary found in legislative texts. Input legal documents were tokenized with a maximum length of 512 tokens, while target summaries were limited to 64 tokens, striking a balance between content coverage and computational efficiency.

The fine-tuning process was conducted using Google Colab's GPU resources, which provided sufficient computational power for our model development and experimentation. While we initially considered larger-scale fine-tuning on more powerful infrastructure such as Kaggle's A100 GPU environment or MSU HPCC clusters, the Colab environment proved adequate for our proof-of-concept implementation and allowed for rapid iteration during the development phase.

Our fine-tuned model for legal and policy text summarization has been successfully deployed and is publicly available on Hugging Face at `AtharvaKirk/legal-summarizer-distilbart`. This deployment makes our specialized legal summarization capabilities accessible to researchers and practitioners in the legal technology domain.

4.7 Tokenization

To generate abstractive summaries, we used the fine-tuned DistilBART model and tokenizer. Sentence tokenization was handled using NLTK's Punkt tokenizer.

Token-aware chunking of long documents: Since legal and government documents exceed the model's maximum input length, we implemented a token-based chunking function:

- The input text is first segmented into sentences using NLTK.
- Sentences are iteratively added to a running chunk while tracking their tokenized length (using the DistilBART tokenizer).
- When adding the next sentence would exceed a predefined token limit, the current chunk is closed and a new one is started.
- Empty or non-string inputs are safely ignored, returning an empty list.

This procedure produced a sequence of overlapping-text chunks that are in-line with the model's maximum token capacity, and preserve sentence boundaries and coherence.

Chunk-level summarization: The chunk summarisation function applied the fine-tuned DistilBART model to a single chunk:

- The chunk text is tokenized with truncation to a maximum input length.
- Using greedy beam search, the model generates a summary subject to minimum and maximum target lengths, which are decoded back into text with special tokens removed.
- Additional decoding constraints (length penalty, no repeat ngram size) are added to aid fluent, non-redundant summaries.
- Chunks that are empty or contain only whitespace are skipped to avoid unnecessary computation.

Hierarchical summarisation of full documents: The final function implemented document-level summarisation using a two-stage, hierarchical approach:

1. **Chunking:**

   The full document is first split into token-capped chunks using the token-aware chunking function. A max chunk parameter truncates extremely long inputs for efficiency.

2. **Single-stage case:**

   If the document produces only one chunk, we directly call the chunk summarisation function with final length parameters (summary minimum and maximum lengths) to obtain the document summary.

3. **Two-stage case:**

   - Each chunk is summarised individually with shorter lengths, producing a list of intermediate chunk summaries.
   - To emphasize the most important segments of the document (often containing conclusions or key findings), only the last 10 chunk summaries are concatenated into a meta-input.
   - This meta-input is then summarised again with longer final length constraints, delivering a coherent, high-level document summary.

This hierarchical procedure allowed the model to handle very long reports while still producing an abstractive summary constrained by the model's token limitations. Without this approach, the generated summaries included information only found at the beginning of the document, which most often excluded significant information.

Dataset evaluation with ROUGE: For empirical evaluation, we ran the full summarisation pipeline on a subset of the test data using fixed hyperparameters:

- `chunk_max_tokens=480`,
- `max_chunks=20`, `chunk_min_len=150`,
- `chunk_max_len=250`,
- `final_min_len=200`,
- `final_max_len=350`, `num_beams=6`,
- `max_input_length=512`,
- `length_penalty=0.9`,
- `no_repeat_ngram_size=4`, and
- `early_stopping=True`.

The resulting ROUGE scores quantify how well the fine-tuned DistilBART model preserves key content compared to human-written summaries and can be directly compared against the earlier TF-IDF cosine baseline reported in Table 3 below.

4.8 Chatbot Integration: Retrieval and Generation

User Query: Users can input a question or topic in natural language (e.g., "tenant eviction laws"). The chatbot interface is designed to accept open-ended queries related to legal or policy topics.
Chunking for QA: BillSum dataset have shorter but segregated by sections. The dataset is first split by sections, longer sections are then chunked into smaller chunks using T5 tokenizer. GovReport does not have sections and the long text hold contexts that T5 cannot capture, so a LED tokenizer was used for this dataset.
Retrieval: The system computes SentenceTransformer embeddings for all preprocessed summaries in the database. When a query is entered, cosine similarity is calculated between the query and each summary vector to identify the most semantically relevant documents.
QA Model: The retrieved chunks then pass through the question-answering model (`distilbert-base-uncased-distilled-squad`) to retrieve the best chunk, which passed through the summarizer earlier created. **Output:** The retrieved summary is displayed directly to the user. This retrieval setup aims to pinpoint the exact point within the datasets to the summarizer, ensuring that the responses remain factual and consistent with the original data sources.

Advantages:

- Provides instant responses, making it ideal for lightweight environments such as Kaggle or Colab free-tier sessions.
- Stable and reproducible results, since outputs are based on fixed summaries rather than variable generative outputs.

Note: While a Retrieval Augmented Generation system could produce more flexible, generated summaries, it would also require significantly higher computational resources and fine-tuned abstractive models. The current retrieval-only design ensures feasibility for the midterm evaluation and still demonstrates how summarization can support accessible legal information retrieval. In future work, we plan to incorporate RAG or hybrid models with an instruct-based generator if sufficient GPU resources become available, enabling more context-aware and conversational responses.

5 Results and Evaluation

We employed the ROUGE metric to evaluate both our extractive baseline and fine-tuned abstractive models against human-written reference summaries. ROUGE measures the degree of n-gram overlap between generated and reference summaries, providing a quantitative estimate of content preservation and summary quality.

Model	R-1	R-2	R-L
TF-IDF Baseline	0.27	0.12	0.18

Table 2: Extractive ROUGE scores on validation data

Table 2 presents the ROUGE scores for our initial TF-IDF cosine similarity baseline model, which established a performance floor for extractive methods on legal documents. The baseline demonstrates moderate recall of key terms but exhibits expected limitations of sentence-level extractive approaches, including lack of abstraction, limited paraphrasing capability, and reduced overall fluency.

Model	R-1	R-2	R-L
TF-IDF	0.27	0.12	0.18
DistilBART	0.42	0.23	0.38

Table 3: Comparative ROUGE scores: baseline vs. fine-tuned

Table 3 shows that our fine-tuned DistilBART model exceeded our target performance objectives, achieving ROUGE-1 above 0.40, ROUGE-2 above 0.20, and ROUGE-L above 0.35. These results demonstrate enhanced content coverage and summary coherence compared to the extractive baseline.

The performance improvement from baseline to fine-tuned abstractive model represents a 55.6% increase in ROUGE-1, 91.7% increase in ROUGE-2, and 111.1% increase in ROUGE-L scores. These substantial gains validate our approach of applying transformer-based abstractive models to legal document summarization and demonstrate the limitations of purely extractive methods for this domain.

These scores are considered strong and acceptable for legal-domain summarization and QA according to two credible sources:

- **Kornilova & Eidelman (2019)** – BillSum corpus creators, providing ROUGE-1 performance expectations for legislative text summarization to be acceptable when greater than 0.40.
- **Shukla et al. (2022)** – Legal document summarization analysis establishing ROUGE-2 benchmarks for evaluating bigram-level content preservation to be acceptable when greater than 0.20.

What do these scores mean?

- **ROUGE-1**: Measures the percentage of key legal terminology from the reference summaries that the model successfully captures.
- **ROUGE-2**: Evaluates how many important legal phrase pairs (bigrams) are preserved, reflecting the model's ability to retain multi-word legal concepts.
- **ROUGE-L**: Assesses sentence-level structural similarity to human-written summaries, indicating how well the model preserves the overall organization and flow of the legal text.

The achieved ROUGE scores align with performance levels reported in legal summarization literature and confirm that our fine-tuned DistilBART model successfully generates more concise, semantically rich summaries while maintaining factual accuracy and legal relevance. The hierarchical processing approach effectively handles long legal documents while preserving contextual coherence across document sections.

6 Interface Design and Implementation

We developed an intuitive web-based interface using Gradio to make our legal summarization system accessible to non-technical users. The interface

features two primary modes of interaction to accommodate different user needs.

The first mode allows users to input raw legal text directly into a text area, which is then processed by our fine-tuned DistilBART model to generate a concise summary. This functionality is particularly valuable for legal professionals, students, and journalists who need to quickly understand lengthy legal documents.

The second mode provides a retrieval-based question answering system, where users can pose natural language questions about legal topics. The system retrieves relevant information from our processed legal corpus and presents synthesized answers, enabling users to access specific legal information without navigating entire documents.

The interface design emphasizes simplicity and usability, with clear input fields, responsive feedback, and well-formatted output sections. This dual-mode approach ensures that our system can serve both comprehensive document analysis and targeted information retrieval needs, making legal document analysis more accessible to a broader audience.

7 Conclusions and Future Work

This work presents a comprehensive NLP framework for legal and policy document summarization that addresses the critical challenge of making complex legislative and regulatory materials accessible to non-experts. Our approach combines extractive and abstractive summarization techniques with a retrieval-based chatbot interface, enabling users to efficiently navigate and understand lengthy legal documents.

Our fine-tuned DistilBART model achieved substantial improvements over the TF-IDF baseline: 55.6% increase in ROUGE-1, 91.7% in ROUGE-2, and 111.1% in ROUGE-L scores. The hierarchical chunking approach successfully handles documents exceeding standard token limits while preserving semantic coherence. Specialized decoding strategies ensure summaries are comprehensive and appropriately detailed or legal text, while maintaining critical elements such as definition, provisions, and temporal constraints.

The system, deployed on Hugging Face as `AtharvaKirk/legal-summarizer-distilbart`, proves it can be used effectively through its retrieval-only design that ensures fast responses without requiring extensive GPU. We created a web interface using Gradio that allows users to input lengthy bills and receive comprehensive summaries, as well as ask specific questions about bills to get targeted answers identifying which sections are relevant to their queries. By including both BillSum and GovReport datasets, our framework can handle various types of legal documents, making legal information more accessible to anyone who needs it.

For future work, there are several directions would enhance the framework's capabilities. Integrating hybrid models would enable more context-aware responses when sufficient GPU resources are available, allowing the system to synthesize information across multiple documents. Incorporating broader datasets, including judicial opinions, regulatory guidance, international treaties, would broaden the system's applicability to a wider range of legal materials. Additionally, multilingual support and compliance checking features would extend the framework's targets and practical application. These development would significantly enhance legal information access and transparency, supporting civic participation and more efficient legal research.

Bios and Responsibilities

Atharva Kirkole (kirkolea), Master's First Year, Computer Science Engineering

Responsibility: Implemented baseline NLP models and fine-tuned models on legal/policy datasets for summarisation. Interpreted model performance and how model outputs can support better decision-making, compliance checking, or law review tasks. Developed UI for project demo.

Yuxuan Zhi (zhiyuxua), First Year PhD student in Educational Psychology

Responsibility: Conducted research on industry benchmarks for law and policy summarisation tasks. Led the setup of the reporting framework and prepared written reports and presentation slides for the project.

Mahnoor Sheikh (sheikhm6), Master's Second Year, Data Science

Responsibility: Collected and cleaned legal/policy text datasets. Handled preprocessing tasks like sentence tokenization, and formatting documents into model-ready inputs. Incorporated chunking in abstractive summarisation model to improve results. Set up UI for project demo.

Vaibhav Vennam (vennamva), Master's Second Year, Data Science

Responsibility: Developed the QA Retrieval/RAG model, and created the legal-to-structured output format for the user chatbot.

References

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. 2021. Automatic summarization of legal bills: A comparative analysis of classical extractive approaches. In *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pages 394–400.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. 2022. Legal case document summarization: Extractive and abstractive methods and their evaluation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1048–1064, Online only. Association for Computational Linguistics.