Enhancing Legal Document Summarization for Professionals: An Extractive Approach

Manasvi Kalyan
SCSET
Bennett University
Greater Noida, India
e21cseu0405@bennett.edu.in

Pratik Dwivedi
SCSET
Bennett University
Greater Noida, India
e21cseu0096@bennett.edu.in

***Abstract*—The increasing volume and complexity of legal documents pose significant challenges for legal professionals who must extract relevant information efficiently. This paper addresses this issue by focusing on extractive summarization of legal documents, a technique that selects key sentences to create concise summaries while retaining the original language and structure. We utilize a labeled dataset of legal judgments and their summaries, employing both binary and multi-class labeling methods to categorize sentences based on their summary-worthiness and segment type (Facts, Arguments, Issue, Analysis, or None). To enhance our dataset, we convert abstractive summaries into extractive formats and fine-tune our models using a combination of pre-existing datasets and our labeled data. We train and evaluate various models, including traditional machine learning algorithms, neural networks, and transformer-based models, using metrics such as precision, recall, F1-score, ROUGE, and BERTScore. Our findings highlight the challenges posed by class imbalance and the need for domain-specific adaptations in legal document summarization. The proposed methodology aims to improve the efficiency and accuracy of processing legal texts, ultimately aiding legal practitioners in their work.**

I. Introduction

In the domain of legal practice, the volume and complexity of legal documents present significant challenges. These documents are often lengthy, densely packed with technical language, and require substantial time to read and comprehend. This complexity makes it difficult for legal professionals, researchers, and other stakeholders to extract relevant information efficiently, potentially leading to delays and misinterpretations.[11][8]

To address this challenge, we focus on extractive legal document summarization, a technique that selects key sentences from a document to create a concise summary.[7][12] We opted for extractive summarization over abstractive summarization for several reasons. First, extractive summarization retains the original structure and language of the source material, preserving the technical precision inherent in legal documents. This approach is particularly valuable in legal contexts, where exact wording can be critical. Second, extractive summarization requires less complex training, as it does not involve generating new text but rather selecting from existing sentences.[7][8] This simplicity in training reduces computational costs and the need to learn extensive vocabulary or intricate text relationships.

In our study, we use a labelled dataset comprising legal judgments and their corresponding summaries.[16] This dataset allows us to generate numeric labels for each sentence, indicating its summary-worthiness. By leveraging these labels, we can train a classifier that predicts which sentences in a legal document should be included in a summary. This method provides a structured approach to extractive summarization[9], enabling more efficient and accurate processing of legal documents.

The following sections of this paper detail our methodology, dataset preparation, and experimental results. We conclude with a discussion of the implications of our findings and potential future directions for enhancing extractive legal document summarization.

II. Literature Review

Legal judgment documents, particularly those from Common Law systems like India, the UK, and the USA, are often extensive and intricate. These documents contain a wealth of information that is crucial for legal practitioners. However, the sheer volume and complexity of these documents pose a significant challenge. Legal practitioners often must sift through hundreds, if not thousands, of case judgments to identify relevant cases that they can cite as precedents in an ongoing case. This process of identifying and understanding relevant information from legal judgment documents is not only time-consuming but also mentally exhausting. It requires a high level of expertise and attention to detail, as missing out on key information could potentially impact the outcome of a case. Given the increasing number of legal cases and the growing size of legal judgment documents, this task is becoming even more daunting and impractical. This necessitates the automation of the task.[13]

Automatic text summarization, particularly extractive summarization, offers a promising solution to this problem. Extractive summarization aims to generate a concise and coherent summary by extracting the most important sentences or phrases from the original document.[14][8][7] This approach maintains the original phrasing and context, making it particularly suitable for legal judgment documents, which often contain complex legal terms and arguments, with many existing methods for text summarization, there is a lack of detailed analysis on how these methods perform on legal case documents.[15][6]

However, the application of automatic text summarization in the context of legal judgment documents is still in its early stages. There are several challenges that need to be addressed,[13] including the need for better understanding of legal context, improving the coherence of the generated summaries,[3] and dealing with the length and complexity of legal judgment documents. A plethora of solutions exists for text summarization, including extractive and abstractive, supervised, and unsupervised methods.[10][2] However, detailed systematic analyses are rare on how the different families of summarization models perform on legal case documents.[5] Research by Abhay Shukla[16] took an early step in this direction, but it mostly considered extractive methods. The state-of-the-art in document summarization has advanced rapidly in the last couple of years, and there has not been much exploration on how recent transformer-based summarization models perform on legal documents.

Although a recent research paper [5] took initiative to adapt state-of-the-art NLP models such as BERT and its variants to legal domain, since they were underperforming in specialized domains like it, its distinct characteristics such as specialized vocabulary and particularly formal syntax, presents unique challenges for these models. To address these challenges, researchers have proposed further pre-training BERT on domain-specific corpora or pre-training BERT from scratch on domain-specific corpora.[5][4] These strategies have been systematically explored in the legal domain, with promising results. Further pre-training or pre-training BERT from scratch on domain-specific corpora has been found to perform better than using BERT out of the box for domain-specific tasks. They released LEGAL-BERT, a family of BERT models for the legal domain, intended to assist legal NLP research, computational law, and legal technology applications.

A thorough analysis of state-of-the-art text summarization approaches, which build on recent advances in deep learning and language models and compared their basic techniques [9]. Evaluated the performance of these approaches using popular evaluation scores, such as ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-LSUM, BLEU-1, BLEU-2, and SACREBLEU, while comparing them to the best performing extractive approaches (as a baseline).

In conclusion, the literature underscores the need for effective summarization of legal judgment documents.[10] While models like BERT have made

Fig. 1. IN-EXT dataset directory structure

```
├─ judgment
├─ summary
│  ├─ full
│  │  ├─ A1
│  │  └─ A2
│  └─ segment-wise
│     ├─ A1
│     └─ A2
└─ IN-EXT-length.txt
```

significant strides, unique challenges in the legal domain necessitate further research.[5] This paper aims to explore models that can perform text classification, label each sentence in a document, and combine them into a summary. Goal is to understand the needs of a model that enable it to make predictions that enhance the effectiveness of summarization, contributing to making legal documents more accessible and understandable, aiding legal practitioners in their work.

III. Dataset

Figure 1 describes the structure of the IN-EXT dataset used in our study.

We work with a dataset of 50 legal documents, each accompanied by a summary annotated by two different annotators, A1 and A2. The summaries are structured into five key segments: Facts, Arguments, Issue, and Analysis and Statutes. This segmentation allows us to capture the key elements typically found in legal documents and understand their distribution within the judgments. The dataset was introduced and made available from the work done by [16].

Table I presents an overview of three different datasets available, highlighting their language, domain, number of documents, and the average number of tokens in both the documents and their summaries. The first dataset, IN-Ext, comprises 50 Indian court rulings with extractive summaries, featuring an average of 5,389 tokens per document and 1,670 tokens per summary. The second dataset, IN-Abs, includes 7,130 Indian court rulings with abstractive summaries, with an average of 4,378 tokens per document and 1,051 tokens per summary. The third dataset, UK-Abs, consists of 793 UK court rulings with abstractive summaries, averaging 14,296 tokens per document and 1,573 tokens per summary.

TABLE I
DATASET STATISTICS

Dataset	Language	Domain	Doc	Avg # Tokens	
				Doc	**Summ**
IN-Ext (Indian docs, extractive summ)	EN	Court Rulings	50	5,389	1,670
IN-Abs (Indian docs, abstractive summ)	EN	Court Rulings	7,130	4,378	1,051
UK-Abs (UK docs, abstractive summ)	EN	Court Rulings	793	14,296	1,573

IV. Methodology

In this section, we describe our methodology, which includes sentence labeling, dataset preparation, model training, and evaluation metrics. We also discuss the generation of summaries and the evaluation of their quality.

A. Sentence Labeling

To effectively summarize legal documents, we first label the sentences within these documents using two distinct methods: binary labeling and multiclass labeling.[9] To achieve this, we utilize the all-MiniLM-L6-v2 model from the Sentence Transformers library available on huggingface[1] to embed each sentence in the summary and the actual judgment. We then compare the cosine similarity between the embeddings of each sentence pair to determine if they are similar. If the cosine similarity exceeds a predefined threshold of 0.8, we consider the sentences to be the same, and assign the sentence an appropriate label.

Following this comparison, we apply two labeling methods:

1) **Binary Labeling:** In this method, sentences are labeled as either included 1 or not included 0 in the summary based on their similarity to sentences in the summary.This approach simplifies the task by focusing solely on the presence or absence of a sentence in the summary.
2) **Multiclass Labeling:** Here, we categorize sentences into one of five classes: Facts, Issue, Analysis, Argument, and None (indicating it is not part of the summary). This method provides a more granular understanding of the role each sentence plays within the document.

B. Model Training and Preparation

Using the labeled datasets, we prepare each sentence with embeddings and attention masks generated by the InLegalBERT model from Hugging Face. These embeddings provide a rich semantic representation of the sentences, crucial for effective classification.

We train classifier models for both labeling methods, experimenting with various architectures which are listed in II.

TABLE II
MODELS USED FOR EXTRACTIVE SUMMARIZATION

Category	Model 1	Model 2
ML Algorithms	Random Forest Classifier (RFC)	Support Vector Machine (SVM)
Neural Networks	Fully Connected Neural Network (FCNN)	Long Short Term Memory (LSTM)
Transformers	BERT	RoBERTa
Custom Models	SummaRuNNer	BiLSTM with Attention

C. Evaluation Metrics

To evaluate the performance of our classifier models[8], we use several standard metrics:

- **Precision:** Indicates the proportion of true positives among the predicted positives.
- **Recall:** Measures the proportion of true positives among the actual positives.
- **F1 Score:** The harmonic mean of precision and recall, providing a balanced evaluation metric.

Once the classifier model is trained, we generate summaries by selecting sentences labeled as worthy of inclusion. The steps involved are as follows:

1) **Classification:** Each sentence in a legal document is labeled using the trained classifier.
2) **Selection:** Sentences labeled as relevant (e.g., Facts, Issue, Analysis, Argument) are selected and concatenated to form the summary.
3) **Evaluation:** The generated summaries are evaluated using two primary metrics:
   - **ROUGE Score:** Measures the overlap of n-grams between the generated summary and the reference summary, providing a quantitative assessment of content similarity.
   - **BERTScore:** Uses BERT embeddings to compare the semantic similarity between the generated summary and the reference summary, offering a more nuanced evaluation of summary quality.

Through this methodology, we aim to develop a robust extractive summarization model that efficiently processes legal documents, facilitating quicker and more accurate information extraction for legal professionals.

V. EXPERIMENTAL SETUP AND RESULTS

In this section, we present the experimental setup and results of our study. We evaluate the performance of various models for extractive summarization of legal documents using two distinct labeling methods: binary labeling and multi-class labeling. The results are analyzed to understand the strengths and weaknesses of each approach.

A. Binary Labeling Method

In the binary labeling method, sentences are labeled as either included (1) or not included (0) in the summary. This method simplifies the task by focusing solely on the presence or absence of a sentence in the summary.

Table III presents the performance of various models in classifying sentences as either included (1) or not included (0) in the summary. The models

TABLE III
CLASSIFICATION RESULTS FOR BINARY LABELING METHOD

Model	Class	Precision	Recall	F1
RFC	0	0.74	1.00	0.85
	1	1.00	0.01	0.01
SVM	0	0.74	0.99	0.85
	1	0.56	0.04	0.08
FCNN	0	0.78	0.86	0.82
	1	0.39	0.26	0.31
BiLSTM	0	0.76	1.00	0.86
	1	0.27	0.01	0.01
BERT	0	0.83	0.80	0.81
	1	0.42	0.46	0.44
RoBERTa	0	0.76	0.78	0.77
	1	0.38	0.40	0.39
SummaRuNNer	0	0.72	0.46	0.56
	1	0.01	0.00	0.01
BiLSTM Attention	0	0.70	0.43	0.53
	1	0.38	0.22	0.27

demonstrate high precision and recall for the '0' class, indicating that they are effective at identifying sentences that should not be included in the summary but that could also be done by just guessing one label all the time and getting a good score on the test. As suspected, the performance significantly drops for the '1' class, with low precision, recall, and F1-scores across all models. For instance, the Random Forest Classifier (RFC) achieves a precision of 1.00 for the '1' class but with a recall of only 0.01, resulting in an F1-score of 0.01, suggesting that it rarely correctly identifies included sentences.

However, the recall for BERT was 0.80 and 0.46 for class '0' and '1' respectively, similarly RoBERTa had recall of 0.78 and 0.40 for class '0' and '1' respectively, indicating that more relevant sentences were identified compared to other models. This suggests that while the models other than the pre-trained transformers are accurate when they do make positive predictions, they miss a substantial number of sentences that should be included in the summary if they are trained from scratch without the understanding of the nuances of the language. This issue is exacerbated by the imbalanced nature of the dataset, where non-summary sentences far outnumber summary sentences which only hinders the model in its ability to see a significant amount of data for each class equally to understand it well enough.

B. Transition to Multi-Class Labeling Method

Recognizing the limitations of binary labeling, we explored a more nuanced approach with multi-class labeling. In this method, sentences are categorized into one of five classes: Facts, Issue, Analysis, Argument, and None. we can incorporate more detailed labels that better represent the varied importance of sentences, which is expected to improve the models' ability to identify sentences that should be included in the summary. This approach can help in capturing subtler distinctions and relationships in the data, potentially leading to better overall performance..

C. Multi-Class Labeling Method

Table IV presents the performance of various models using the multi-class labeling method, one thing to note here is the implementation of the SummaRuNNer model, implemented with our understanding of its nature, it is still a naive approach that combines the embeddings of the entire document and the embedding of one sentence at a time to classify the sentence into a category (binary or multi class), further improvements can be expected in this model if combined with the rich representation returned buy the BERT models.

Table IV presents the performance of various models using the multi-class labeling method. The results show a wide range of performance across different models and classes. Notably, models such as BERT and RoBERTa exhibit superior performance, especially for the none class, with BERT achieving an F1 score of 0.88 and RoBERTa 0.77. These models benefit from extensive pretraining on large, diverse text corpora, allowing them to capture rich contextual information and nuanced language patterns. Consequently, they can better generalize to different classes in the multi-class labeling task.

On the other hand, models like RFC, SVM, and FCNN struggle significantly with precision and recall for the analysis, argument, and facts classes, often achieving scores close to zero. This underperformance can be attributed to their relatively simple architectures and limited capacity to capture complex semantic relationships within the

TABLE IV
CLASSIFICATION RESULTS FOR MULTI-CLASS LABELING METHOD

Model	Class	Precision	Recall	F1
RFC	analysis	0.00	0.00	0.00
	argument	0.00	0.00	0.00
	facts	0.00	0.00	0.00
	judgement	0.50	0.21	0.30
	none	0.78	1.00	0.88
SVM	analysis	0.35	0.05	0.05
	argument	0.33	0.3	0.06
	facts	0.78	0.06	0.12
	judgement	0.48	0.43	0.45
	none	0.79	0.97	0.88
FCNN	analysis	0.22	0.18	0.20
	argument	0.12	0.12	0.12
	facts	0.35	0.14	0.20
	judgement	0.39	0.39	0.39
	none	0.81	0.87	0.84
BiLSTM	analysis	0.28	0.24	0.25
	argument	0.14	0.19	0.16
	facts	0.31	0.24	0.27
	judgement	0.47	0.54	0.50
	none	0.82	0.84	0.83
BERT	analysis	0.00	0.00	0.00
	argument	0.00	0.00	0.00
	facts	0.53	0.19	0.28
	judgement	0.69	0.32	0.44
	none	0.79	0.98	0.88
RoBERTa	analysis	0.00	0.00	0.00
	argument	0.00	0.00	0.00
	facts	0.44	0.19	0.26
	judgement	0.64	0.32	0.43
	none	0.72	0.92	0.77
SummaRuNNer	analysis	0.25	0.20	0.22
	argument	0.10	0.05	0.07
	facts	0.40	0.25	0.31
	judgement	0.50	0.15	0.23
	none	0.75	0.85	0.80
BiLSTM Attention	analysis	0.30	0.25	0.27
	argument	0.20	0.15	0.17
	facts	0.50	0.20	0.28
	judgement	0.60	0.30	0.40
	none	0.78	0.88	0.83

text. SummaRuNNer, while showing a balanced performance for analysis, facts, and none, falls behind the BERT models but indicates potential for improvement. Enhancing SummaRuNNer with more sophisticated embeddings, such as those from BERT, could significantly boost its performance. The BiLSTM with Attention model also shows moderate performance, with better results than simpler models but still lagging behind the pretrained models. This highlights the importance of high-quality embeddings and the benefits of leveraging pretrained language models for complex multi-class labeling tasks.

The multi-class labeling method posed additional challenges, as models had to differentiate between various types of sentences. BERT and RoBERTa again outperformed other models, with BERT achieving a precision of 0.87 and an F1-score of 0.47. However, the overall performance metrics were lower compared to the binary labeling method due to the increased complexity of the task.

A notable observation is the reduced recall across all models, further highlighting the difficulty in identifying and correctly classifying each sentence's role. For example, the recall for RoBERTa was 0.31, indicating that the model struggled to capture all relevant sentences within their appropriate categories.

TABLE V
SUPPORT VALUES BINARY CLASSIFICATION IN TEST SET

Labels	**Support**
0	985
1	319

TABLE VI
SUPPORT VALUES MULTI-CLASS CLASSIFICATION IN TEST SET

Labels	**Support**
analysis	204
argument	32
facts	112
judgement	28
none	1356

The tables VI and V represent the support values in the classification results highlight the distribution of instances across different classes for both binary and multiclass classification and provide crucial insight into the impact of class imbalance on model performance. In both binary and multi-class classification tasks, the support values reveal significant imbalances, with some classes having substantially more instances than others. For instance, in the multi-class classification, class 4 has 1356 instances, whereas class 1 has only 32 instances. Similarly, in binary classification, class 0 has 985 instances compared to 319 instances in class 1. This imbalance can lead to models being biased towards the majority class, achieving high precision and recall for the well-represented class while underperforming on the minority classes. The imbalance during training affects the model's ability to generalize well across all classes, often resulting in lower precision, recall, and F1 scores for underrepresented classes. Addressing this imbalance through techniques like oversampling, undersampling, or using class weighting during training could improve the overall performance and ensure a more balanced evaluation across all classes.

VI. Conclusion

The comparison of binary and multi-class labeling methods provides valuable insights into the performance of various models in summarization tasks. In the binary labeling method, where sentences are simply labeled as included or not included in the summary, models like RFC and SVM exhibit high precision for the 0 class but struggle significantly with the 1 class, leading to poor recall and F1 scores. This suggests that these models are heavily biased towards predicting the majority class, resulting in an imbalanced performance, due to the imbalanced nature and quantity of the dataset. BERT and RoBERTa, despite being pretrained on large datasets, also show limitations in capturing the nuances required for accurate binary classification, highlighting the challenges inherent in this simplistic labeling approach.

In contrast, the multi-class labeling method offers a more granular evaluation of model performance across different categories such as analysis, argument, facts, judgement, and none. Here, models like BERT and RoBERTa shine due to their advanced pretraining, achieving significantly higher precision, recall, and F1 scores across multiple classes, yet still struggle with some classes as if they have know knowledge of them. However, simpler models like RFC and SVM continue to underperform, particularly for more complex classes, indicating their inability to effectively handle the intricacies of multi-class text classification. SummaRuNNer, although a more naive approach, demonstrates potential with a balanced performance that could be enhanced by integrating richer embeddings from pretrained models like BERT.

The findings underscore the importance of domain-specific pretraining and the need for more comprehensive datasets. Models benefit significantly from large-scale pretraining, but their performance can be further improved with additional domain-specific data that helps capture the unique language patterns and context of the target domain. Our results suggest that while current pretrained models like BERT and RoBERTa are powerful, integrating them with domain-specific adaptations and utilizing high-quality embeddings can lead to more robust and accurate summarization systems. Future work should focus on gathering more domain-specific data and exploring advanced pretraining techniques tailored to the specific requirements of the task at hand.

References

[1] Tom Aarsen. *Sentence-transformers/all-minilm-L6-V2 · hugging face*. URL: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

[2] Narendra Andhale and Laxmi Bewoor. "An overview of Text Summarization techniques". In: Aug. 2016, pp. 1–7. DOI: 10.1109/ICCUBEA.2016.7860024.

[3] Paheli Bhattacharya, Kaustubh Hiware, Subham Rajgaria, et al. "A Comparative Study of Summarization Algorithms Applied to Legal Case Judgments". In: Apr. 2019, pp. 413–428. ISBN: 978-3-030-15711-1. DOI: 10.1007/978-3-030-15712-8_27.

[4] Paheli Bhattacharya, Soham Poddar, Koustav Rudra, et al. "Incorporating domain knowledge for extractive summarization of legal case documents". In: June 2021, pp. 22–31. DOI: 10.1145/3462757.3466092.

[5] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, et al. *LEGAL-BERT: The Muppets straight out of Law School*. 2020. arXiv: 2010.02559 `[cs.CL]`.

[6] Utkarsh Dixit, Sonam Gupta, Arun Kumar Yadav, et al. "Analyzing the Impact of Extractive Summarization Techniques on Legal Text". In: Jan. 2024, pp. 585–602. ISBN: 978-981-99-6543-4. DOI: 10.1007/978-981-99-6544-1_44.

[7] M. Gupta, N. Narayana, V. Charan, et al. "Extractive Summarization of Indian Legal Documents". In: Apr. 2022, pp. 629–638. ISBN: 978-981-19-0018-1. DOI: 10.1007/978-981-19-0019-8_47.

[8] Ben Hachey and Claire Grover. "Extractive summarisation of legal texts". In: *Artificial Intelligence and Law* 14 (2006), pp. 305–345. URL: https://api.semanticscholar.org/CorpusID:10295764.

[9] Deepali Jain, Malaya Borah, and Anupam Biswas. "Summarization of Indian Legal Judgement Documents via Ensembling of Contextual Embedding based MLP Models". In: Dec. 2021.

[10] Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. "Summarization of legal documents: Where are we now and the way forward". In: *Computer Science Review* 40 (2021), p. 100388. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2021.100388. URL: https://www.sciencedirect.com/science/article/pii/S1574013721000289.

[11] Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. "Text summarization from legal documents: a survey". In: *Artificial Intelligence Review* 51 (Mar. 2019). DOI: 10.1007/s10462-017-9566-2.

[12] Rahul Kore, Prachi Ray, Priyanka Lade, et al. "Legal Document Summarization Using Nlp and Ml Techniques". In: *International Journal of Engineering and Computer Science* 9 (May 2020), pp. 25039–25046. DOI: 10.18535/ijecs/v9i05.4488.

[13] Nikita, Dipti P. Rana, and Rupa G. Mehta. "Research Challenges for Legal Document Summarization". In: *2023 IEEE World Conference on Applied Intelligence and Computing (AIC)*. 2023, pp. 307–312. DOI: 10.1109/AIC57670.2023.10263906.

[14] Saloni Sharma, Surabhi Srivastava, Pradeepika Verma, et al. "A Comprehensive Analysis of Indian Legal Documents Summarization Techniques". In: *SN Computer Science* 4 (Aug. 2023). DOI: 10.1007/s42979-023-01983-y.

[15] Saloni Sharma, Surabhi Srivastava, Pradeepika Verma, et al. "A Comprehensive Analysis of Indian Legal Documents Summarization Techniques". In: *SN Computer Science* 4 (Aug. 2023). DOI: 10.1007/s42979-023-01983-y.

[16] Abhay Shukla, Paheli Bhattacharya, Soham Poddar, et al. "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)*. Ed. by Yulan He, Heng Ji, Sujian Li, et al. Online only: Association for Computational Linguistics, Nov. 2022, pp. 1048–1064. URL: https://aclanthology.org/2022.aacl-main.77.