Engineered Science
DOI: https://dx.doi.org/10.30919/es1022

Indian Legal Corpus (ILC): A Dataset for A dataset summarizing Indian Legal Proceedings using Natural Language

Pawan Trivedi,[1] Digha Jain,[2] Shilpa Gite,[2, 3,*] Ketan Kotecha,[3] Anant Bhatt[4] and Nithesh Naik[5]

Abstract

There is a significant backlog of legal proceedings in several large countries, including India. Technological advancements have been made in intelligent devices that can process and summarize legal documents. However, developing such data-driven systems requires a scarcity of high-quality corpora. Legal AI uses artificial intelligence technology, particularly Natural Language Processing (NLP), to help with legal duties. Legal professionals frequently consider how to solve problems using rule-and symbol-based methods, but NLP researchers are more interested in data-driven and embedding methods. So, in this paper, we present Indian Legal Corpus (ILC), a dataset for Indian legal document summarization. Our dataset differs from the existing summarization datasets in a way that our summaries are highly abstractive. This dataset offers new research opportunities for Legal documents with an abstractive approach. ILC is highly abstractive, concise, and of high quality, as indicated by human and intrinsic evaluation. We are releasing our dataset and models to encourage future research on Legal abstractive summarization.

Keywords: Summarization; Abstractive; Extractive; Legal document summarization.
Received: 31 August 2023; Revised: 11 November 2023; Accepted: 16 November 2023.
Article type: Research article.

1. Introduction

The task of automatically generating a concise summary of the text that concisely gives all the information of the incoming text is a fundamental problem in Natural Language Processing.[1] There are two types of approaches to automatic text summarization: extractive and abstractive.[2] Extractive summarization approaches select top 'n' essential sentences from the text and concatenate them to produce summaries; TextRank and Luhn are some of the extractive summarization algorithms.[3] On the other hand, abstractive summarization methods use Natural Language Generation techniques to generate summaries that capture the salient ideas of the input text document.[4] Although abstractive summaries are more cohesive and concise than extractive summaries, they are more challenging to create due to the nature of the work.

This work introduces the ILC abstractive summarization corpus of Indian legal judgments. We collect properly annotated article-summary pairings using a custom crawler. In conclusion, the following are the significant contributions we bring to this paper:

- We release ILC, a dataset containing case-summary pairs, the first publicly available abstractive summarization dataset for Indian legal judgment.
- The ILC corpus contains 3k+ judgment cases and their parallel summaries.
- We are releasing summarization model checkpoints such as LED and making them publicly available.

1.1 Background

Research into automatic text summarization dates back over 70 years ago. In the 1950s, Luhn presented his paper The Automatic Creation of Literature Abstracts,[5] the first-time automatic summarization caught the scientific community's attention. Since then, there have been a lot of domain-independent and dependent approaches for summarization, such as knowledge-based approaches, statistical approaches, linguistic approaches, rhetorical role approaches, *etc*.[6]

[1] *PES University, Bengaluru, 560085, Karnataka, India.*
[2] *Symbiosis Institute of Technology, Symbiosis International (Deemed University), 412115, Pune, India.*
[3] *Symbiosis Centre for Applied AI (SCAAI), Symbiosis International (Deemed University), 412115, Pune, India.*
[4] *Devang Patel Institute of Advance Technology and Research, Charotar University of Science and Technology, Nadiad Petlad Road, Changa, 388421, Gujarat, India.*
[5] *Department of Mechanical and Industrial Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, Karnataka, India.*
**Email:* shilpa.gite@sitpune.edu.in (S. Gite)

Document Summarization is an active field of research. This field has recently changed rapidly, and substantial progress has been made, from extractive approaches to Sequence to sequence (seq2seq) and recent state-of-the-art (SOTA) transformer modelling in NLP. However, the research has been on developing architecture models on short text or documents (such as news, emails, *etc*.). It needs to pay more attention to long documents (such as Legal Cases, Patient Reports, *etc*.) Specific to legal document summarization, there has been little work done due to limitations in the availability of labelled data because drafting summaries and annotating data is a labour-intensive and expensive task.[7,8] So, in this thesis, the authors have considered the problem of document summarization in two different settings: high resources and low resources. This work seeks to contribute to the existing literature on automatic summarization. Overall, this work is one step closer to data-driven summarization for generating summaries without having large amounts of labelled data in the legal domain. A good performance in the automatic summarization of Legal Cases would allow practitioners to save time and money. Manually drafting summaries is labour-intensive and time-consuming. These automatic summaries are beneficial for lawyers and judges to reference other cases and quick understanding of the cases.

There has been a lot of progress in short document summarization, such as emails, news, *etc*., and the next milestone would be to achieve similar results in lengthy document summarization, such as legal documents.[9,10] Therefore, this thesis focuses on the long document summarization and advancing its literature.[11-15] We identify the architecture for lengthy legal document summarization in high and low resources using domain-independent techniques, which are the central themes of this work.

To the best of our knowledge, this is the first work presenting a systematic survey for legal document summarization from classical to state-of-the-art approaches.
The authors may want to refer to the recent research articles. Creating summaries from legal data such as bills, court decisions, and other legal documents is known as summarization for legal language. Since the beginning of time, legal practitioners such as attorneys and judges have relied on legal experts to make summaries of legal documents for reference to comparable instances, and they rely on legal experts to write headnotes known as summaries.[16] There are 25 High Courts in India,[17] 672 District Courts,[18] and a Supreme Court that publishes legal reports in the public domain. Legal institutions hire legal specialists to prepare summaries since legal notes are lengthy documents. However, this is a time-consuming operation that necessitates substantial human engagement. As a result, automated summarizing of legal papers can assist legal practitioners while drastically decreasing human work.

Saravanan *et al*. (2008)[19] developed a unique approach to using probabilistic graphical models for automated text summarization in the legal sector. Grover *et al*. (2003)[20] presented argumentation roles and divided distinct rhetorical roles into sub-categories. A graph-based extractive technique for the summary of judicial judgments is described by Kim *et al*. (2012).[21] A citation-based summary technique for Legal Texts is presented by Galgani *et al*. (2012).[22] Joshi *et al*. (2019)[23] introduced SummCoder, a novel and unsupervised approach for single document extractive text summarization by constructing sentence scoring and selection strategies based on three sentence matrices that were provided as features to the model, namely, sentence content, sentence novelty, and sentence position in the document. A topic-centered, state-of-the-art, unsupervised text summarization approach that focuses on clustering the most relevant sentences is proposed by Srivastava *et al*. (2022).[24] A detailed overview of various automatic text summarization approaches based on mechanisms used in summary generation is presented by Cajueiro *et al*. (2023).[25] Various deep learning approaches for text summarization by forming labelled data using a similarity index between headnotes and main documents described by Anand *et al*. (2022)[26], especially for Legal texts. With the recent interest in LLMs, much work has been done in analyzing the results and performance of such models. Zhang *et al*. (2023)[27] compared ChatGPT's summarization capabilities and traditional supervised fine-tuning methods. Abdel-Salam *et al*. (2022)[28] presented a paper which gives a detailed study of the performance of BERT-based models in summarization tasks.

1.2 Research purpose

Abstractive text summarization has been revived by the effectiveness of sequence-to-sequence (seq2seq) models over the last decade and recent developments in transformer-based models.[29] Due to a lack of publicly available data, there needs to be more effort in summarizing legal papers using abstract methods because manually creating summaries of legal documents is a labour-intensive and time-consuming procedure.

With ILC, we are providing the community an opportunity to explore the abstractive summarisation paradigm with state-of-the-art models. With ILC, we are open-sourcing variants to LED and BigBird models specific to Indian Legal data.

1.3 Related work

NLP approaches in the legal realm have sparked much interest in recent years.[1-6] Several tasks and models have been proposed through Artificial Intelligence for Legal Assistance[Alia][30], such as Legal Judgment Prediction,[31,32] Legal Summarization,[33] Prior Case Retrieval,[34] Legal Question Answering,[35] Catchphrase Extraction,[36] and Semantic Segmentation.[37] Most of the corpus for legal-NLP tasks such as summarization has been either from other countries such as the USA, Canada, or Australia or, if it's for Indian cases, it has been based on rhetorical roles as shown in Table 1.

BillSum: A Corpus for Automatic Summarization of US

Table 1. Available legal datasets for summarization tasks.

Dataset	Split			Avg. Tokens	Data Reference
	Train	Test	Validation		
BillSum	18,949	3,269	1,237	1382	HuggingFace
Rulingbr	6,373	2,125	2,125	-	GitHub
English Contracts	446	-	-	-	GitHub
Australian Corpus	3890	-	-	-	UCI
Indian Legal Doc.	185	50	30	3176	BUILDNyAI

Legislation, a dataset for summarizing US Congressional and California state laws, was proposed by Kornilova *et al*..[38] The data is separated into 18,949 train bills and 3,269 test banknotes in this abstract data set.

Kalamkar *et al*.[39] proposed a corpus for the Automatic Structuring of Legal Documents, and each part of the document is annotated with a label coming from a list of pre-defined Rhetorical Roles, which can be used for summarization tasks. The corpus contains 265 Indian legal documents annotated with rhetorical roles.

In 2012 Galgani, F *et al*.[40] proposed one of the earliest datasets for summarizing legal documents from the Federal Court of Australia. Corpus contains 2816 cases with citation information, which can be used to summarise cases from the Federal Court of Australia from 2007 to 2009.

Manor *et al*.[41] proposed a dataset of legal text snippets paired with summaries. The dataset contains 446 sets of contract sections with corresponding reference summaries.

Vargas Feijó *et al*.[42] proposed RulingBR: A Summarization dataset for Legal Texts, the corpus contains 10,623 decisions from Supremo Tribunal Federal (STF) the highest court in Brazil, cases dating from 2012 to 2018.

2. Data description

Per the explored literature review, our dataset is the first publicly available Indian Legal judgment data for summarization tasks. It has been scrapped from different websites and concatenated into one dataset ILC. It consists of 3073 cases with their respective cases and summaries.

Figure 1(a) shows the sentence count summary. The maximum sentence count is 243 sentences, and in the Case description, the maximum sentence count is 471 sentences; it can be seen that these counts have a significant difference as compared to the mean, stating that very few judgment cases have relatively more text content. In Fig. 1(b), the same trend is followed for the token count, where the max word count in summary is 4267 compared to the mean of the word count for the same, which is 1/8th of the maximum word count. Similarly, the token count for the case description is 7981. Fig. 2(a) shows sentence count distribution among the 3073 cases and their respective summaries; now clearly visible is the count of cases having a lot of text, which is hardly two to three cases. Similarly, the token distribution for cases and summaries can be seen in Fig. 2(b).

3. Methods

This section contains the data acquisition process and data preprocessing steps. Fig. 3 shows the detailed process of ILC dataset creation.

3.1. Data acquisition

As shown in Fig. 3, Judgement cases from different courts are acquired to obtain variations in the dataset and reduce the bias. Summaries are extracted from websites Primelegal,[43] Briefcased,[44] Lawtimes journal,[45] the Indian Kanoon,[46] and respective High Court websites such as the Supreme Court of India, Delhi High Court, *etc*. Several similarity criteria were used to extract the proper document for each summary during the extraction process. This way, an Indian legal case judgment summary dataset is created by combining 3341 legal case papers and their summaries.

The dataset is stored in the Hugging Face datasets library.

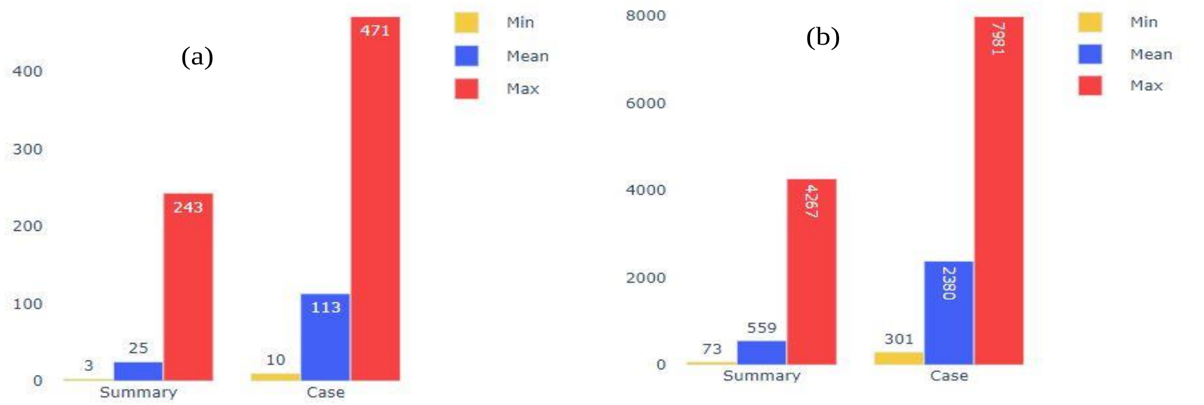

Fig. 1 (a) Sentence Count (b) Token Count.

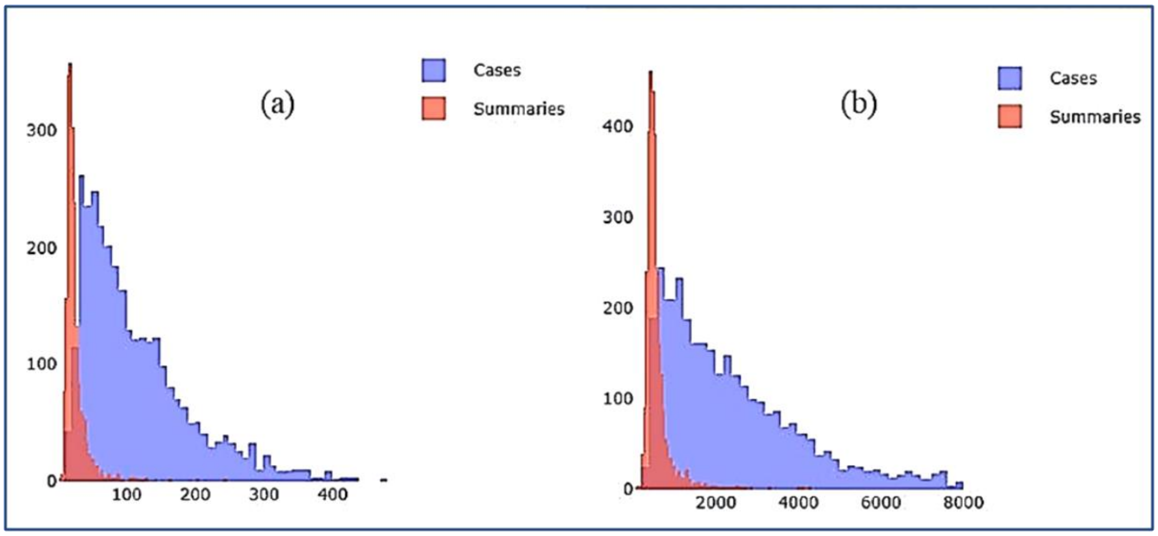

Fig. 2 (a) Sentence Distribution (b) Token Distribution.

Several preprocessing steps have been taken which are discussed in section 3.2 in detail. After extraction, judgment documents are converted from PDF files to .txt format and then both summary and judgment documents are stored in the .txt format and, finally, after preprocessing.

3.2. Data preprocessing

The data preprocessing includes steps to gain essential insights and convert data into a more informative and model-usable format. Our data preprocessing pipeline, as depicted in Fig. 4, included removing extra spaces, blank lines, duplicate words, unreasonable punctuation, and UTF-8 characters. We dropped all the cases with token lengths less than 300 and more than 8000 with their corresponding summaries as they might contain outliers that can affect the performance of the trained model. We also dropped cases where the corresponding summaries token length was less than 70.

The data preprocessing steps we performed on the ILC dataset to make it applicable to the models were as follows:

- URL removal - URLs are removed as they did not add to the information, and to the dataset, it was just a pattern of textual data that had no meaning in itself, hence noise to the model.
- Removal of unnecessary non-alphanumeric characters - There were sequences of characters like “*”, “\\\\”, extra whitespaces, blank lines, *etc*., that are random and are pure noise. Removing these characters not only makes the data more meaningful but also reduces the data size, hence aiding computational time.
- UTF-8 characters' Removal - UTF-8 characters are characters like "\xa0", "\xc2", *etc*., that get into the data when it is web scrapped and converted into text. They are also useless to the model and carry no meaning as text and hence are removed.
- Page no. Removal - Page no. was useless in the summarization tasks and was removed.
- Removal of cases with token lengths less than 300 and more than 8000 - We dropped all the cases with token lengths less than 500 before deleting the punctuations, as they didn't have much information to train the model. We also dropped cases with token lengths of more than 8000 as they might contain outliers that can affect the performance of the trained model.
- There were tokens like “wp253.21.odtwp253.21.odt”,

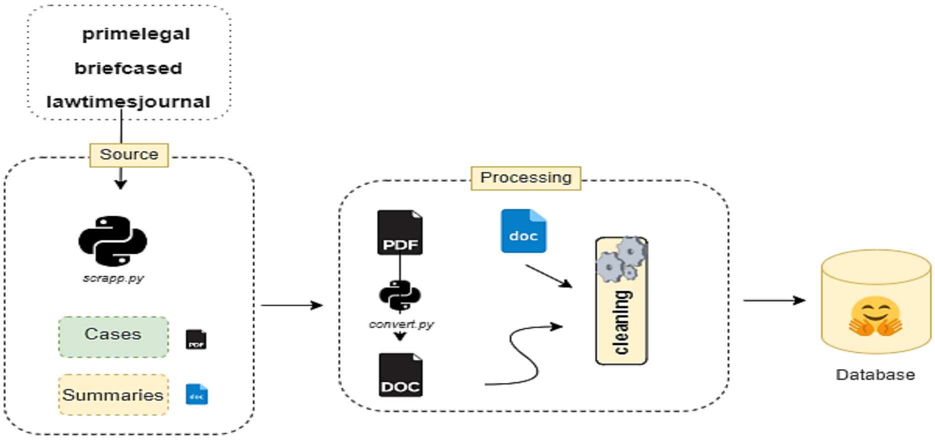

Fig. 3 Data Acquisition system architecture: Three websites, Primelegal, Briefcases and Law Times journal, were web scrapped; the data was first in the form of .doc and .pdf and, later on, were converted into .doc for further preprocessing.

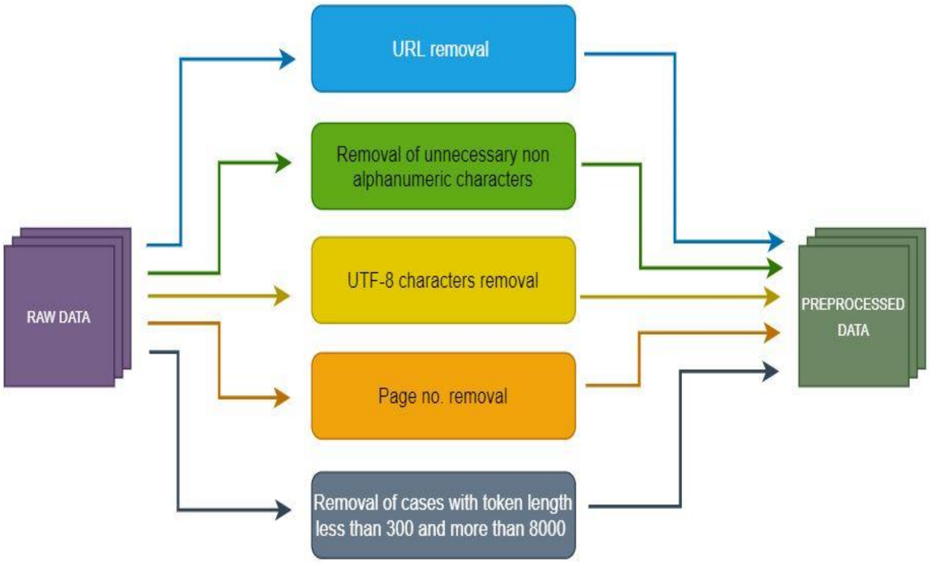
URL removal
Removal of unnecessary non alphanumeric characters
UTF-8 characters removal
Page no. removal
Removal of cases with token length less than 300 and more than 8000
RAW DATA
PREPROCESSED DATA

Fig. 4 Data Preprocessing flow system chart.

“JudgementJudgement", "J U D G E M E N T", "R E P O R T A B L E", "Uploaded", "Downloaded", *etc*. which did not provide any valuable insights and hence were replaced by empty strings.

- There are some prevalent textual data preprocessing steps, like removing stop words and performing stemming and lemmatization over the text, which we have avoided on this dataset for the following reasons:
- Removal of stopwords - Stopwords are words like in, the, which, upon, on, where, *etc*., used to build a sentence, don't carry much information individually and occur in abundance. Stopwords are usually removed in cases like classification, caption/hashtag generation, clustering, *etc*., but in tasks like summarization, language translation, *etc*., where we need a grammatical structure of the sentences, we avoid removing stopwords.
- Stemming and Lemmatization - Stemming converts the word into its base form by removing affixes from it, while in lemmatization, we get the base word with the use of vocabulary and morphological analysis of words. While in stemming, we might get words that do not belong to vocabulary and hence carry no meaning, lemmatization gives words that strictly belong to vocabulary. Stemming and lemmatization can be used in tasks like classification, IR, *etc*. Again, we haven't performed Stemming/Lemmatization on our dataset to get the information like tense and maintain the grammatical structure.

3.3 Data Exploration

In the section, we explored the data and learn more about it. Although the findings in this section won’t be applied anywhere. In this part, our solemn goal was to extract n-grams, where n is in the range.[1,3] We first prepossessed the data by applying the following steps:

- Remove stopwords: We removed stopwords to get more case-specific words.
- Remove common words: We removed the words that were common in many cases so that we get rid of law-related terms and have more case-specific terms with us.
- Lemmatize the words: To get root words in the output and remove repetition of words with the exact root words.
- Then, due to the limitation of RAM, we randomly selected 20 cases out of the ILC dataset, created unigrams and bigrams for them and presented graphs in Figs. 5 and 6.

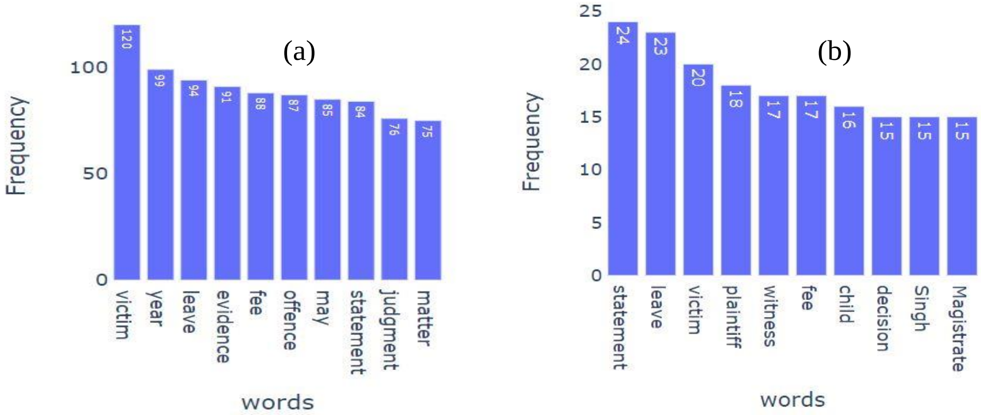

Fig. 5 (a) Unigram Case (b) Unigram Summary.

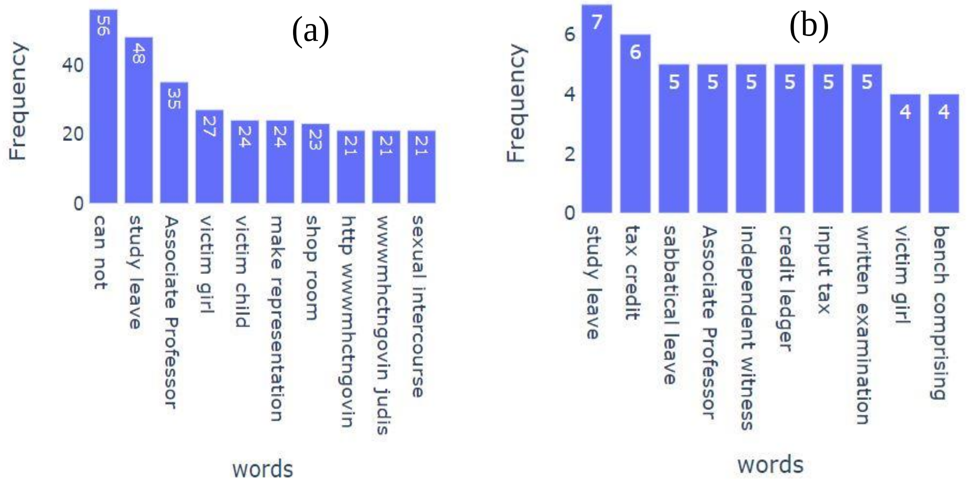

Fig. 6 (a) Bigram Case (b) Bigram Summary.

3.4. Use-Cases of ILC

Currently, there are only a few datasets available for legal document summarization, and ILC differs from these publicly available datasets on the following points:

- Datasets specific to Indian documents, such as judgments, are mainly based on rhetorical role and citation role approaches. ILC is abstractive, as it does not need any specific annotation.
- Other available datasets are from countries such as the USA, Australia, *etc*. Legal documents such as judgments differ a lot from one country to another, so this ILC dataset is the first publicly available dataset that is abstractive.

ILC dataset can be used for both Extractive and Abstractive summarization approaches.

4. Experimental study

We evaluate several extractive and abstractive summarization approaches to establish a benchmark on the ILC dataset. We computed the Rouge-2 Precision and Rouge-2 F score.

4.1 Extractive summarization

Extractive summarizing approaches function by employing a scoring mechanism to identify critical areas of the content and then using selection criteria to pick the k best sentences, which are then concatenated to make a summary. The most essential sentences from the input material are summarized in these summaries. As shown in Fig. 7, every extractive-based approach follows a three-step process:

- Form an intermediate version of the input text that emphasizes the most critical aspects of the text. Topic Representation and Indicator Representation are the two strategies for forming the intermediate representation.
- Based on the representation score, the sentences
- Select a summary comprising some sentences.

The five extractive approaches used in our experiments are described in this section. We used the implementations provided by Sumy.[47]

4.1.1 LexRank

LexRank[39] uses an unsupervised graph-based approach for automatic text summarization. It uses the graph method to score the sentences. LexRank uses eigenvector centrality for computing sentence importance in a graph representation of sentences. The main aim of this approach is to calculate the relative importance of sentences, which is done by a stochastic graph.

This approach uses an adjacency matrix of a graph representation of sentences. This adjacency matrix is the connectivity matrix based on the intra-sentence cosine similarity. In this sentence extraction approach, a sentence that serves as a centroid and the mean for all other phrases is chosen. The sentences are then graded based on their similarities. The LexRank method is as follows:

- Based on Eigen Vector Centrality.
- The vertices of the graphs are where sentences are inserted.
- The weight on the Edges is calculated using the cosine similarity metric.

4.1.2 TextRank

Mihalcea *et al.*[48] proposed a graph-based ranking model, TextRank, and this algorithm uses cosine similarity for forming summaries. Cosine similarity is a technique for estimating the similarity of two word/sentence vectors. In this approach, we use the cosine of the angle between the two vectors to calculate the similarity between two text blocks.

The following approach is for forming summaries using TextRank:

- Develop a Similarity Matrix
- Convert the matrix above to a graph, with nodes representing sentences and edges reflecting similarity scores between sentences.
- Apply the algorithm on the graph to arrive at the sentence rankings.
- Extract the top k sentences for summary generation based on their rankings.

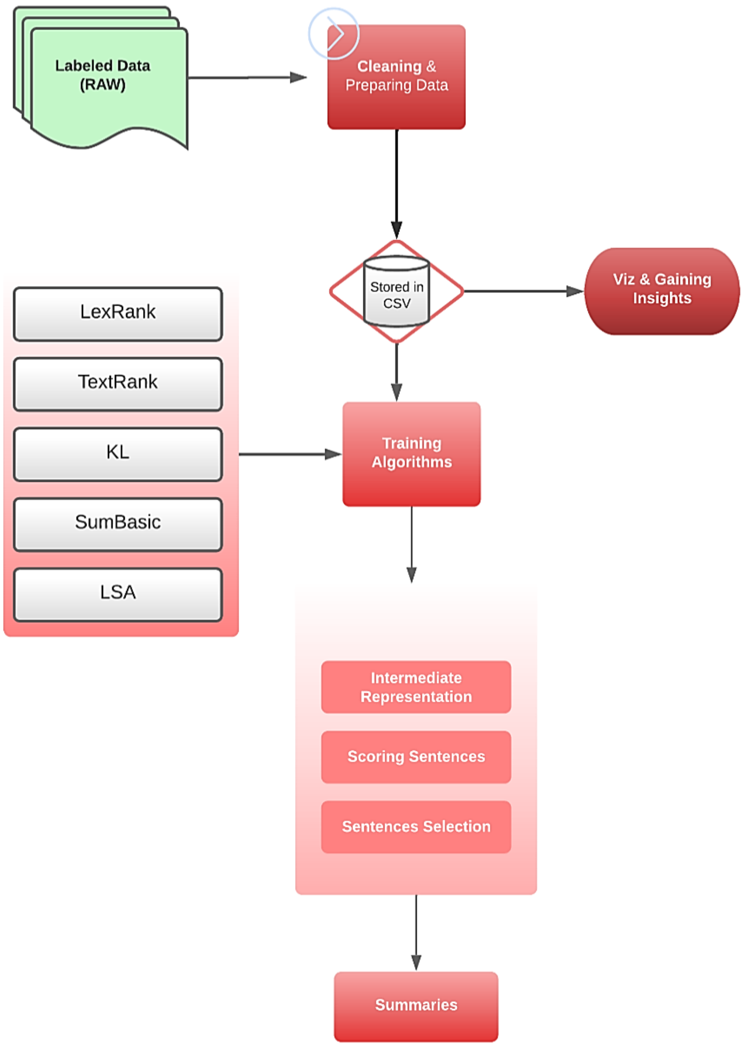

Fig. 7 Our approach of Extractive Summarization.

To calculate cosine similarity, we can calculate the dot product of two vectors: $\frac{a.b}{||a||\times||b||}$. The higher the number of votes cast for a vertex/node, the higher the importance of the vertex.

4.1.3 KLSum

The KL ((Kullback-Lieber)[49] summarizer chooses sentences based on word distribution similarity to the original text. Its goal is to make the KL-divergence criterion more stringent. The Kullback–Leibler divergence is a statistical distance that indicates how far one probability distribution Q is from another. It employs a greedy optimization strategy, adding sentences until the KL-divergence, an entropy measure, falls below a certain threshold. The KL sum approach aims to find a group of sentences with a length of fewer than L words and a unigram distribution that is as close to the original content as possible. The less difference, the more comparable the summary and document are in comprehending ability and meaning delivered.

The difference between two probability distributions is calculated using KL-Divergence. Probability distribution P to an arbitrary probability distribution Q. It compares the unigram probability distributions learned from the observed p(w/R) and new q(w/N) document sets.

4.1.4 SumBasic

Sumbasic[50] is a word frequency approach. To generate a more likely summary in human abstract, Sumbasic utilizes frequently occurring words rather than the less frequent words in a document. Each content word's probability (*i.e.*, numerals, nouns, words, and adj.) is calculated by measuring its frequency in the document collection. Each phrase is graded by considering the average likelihood of the words in it. A simple Greedy Search Algorithm is used to generate the summary. The sentences are chosen iteratively, starting with the highest-scoring content word and breaking ties using the average sentence score. This process is repeated until the maximum length of the summary has been achieved. SumBasic squares the probability of the words in the selected phrase to avoid selecting the same or similar text multiple times, modelling the likelihood of a word occurring twice in summary.

4.1.5 LSA

Latent Semantic Analysis (LSA)[50] extracts hidden semantic structures of words and sentences; it is an algebraic-statistical method. It does not need any training or external knowledge due to the nature of unsupervised learning. LSA gathers information from the context of the input material, such as which standard terms appear in various phrases and which words are used together. LSA chooses the phrases and sentences that most accurately portray the issue.

The basic idea is that the LSA technique forms a matrix of terms and sentences on the rows and columns. The number of times each term appears in each sentence is the value at the intersection of row and column. In text summarization, sentences most similar to others are the most important. The similarity is determined by first reducing the matrix mathematically, and then the cosines of the angles of the vectors in the reduced matrix are compared to find similar rows. LSA uses Singular Value Decomposition (SVD) for matrix Factorization.

The LSA algorithm consists of three steps:

- Input matrix creation - Frequency of the word, Binary Representation, and TF-IDF (Term Frequency-Inverse Document Frequency are three different ways to create an input matrix.
- Singular Value decomposition (SVD)
- Sentence Selection using Topic Methods

4.2. Abstractive summarization

The objective of abstractive text summarizing is to create a summary that captures the main points of the input text material. The term 'abstractive' describes a summary that isn't only a selection of a few existing sections or phrases obtained from the source but a compressed paraphrase of the text's core contents, perhaps using words not found in the source material.

As seen in Fig. 8, this works by pre-training a language model on the unlabeled dataset and then fine-tuning that on downstream tasks, *i.e.* summarization on a specific dataset.

We have used LED and BigBird, two long document transformer models, for abstractive summarization. We have used the HuggingFace[51] library to train these models and make an inference on test data.

4.2.1 Longformer Encoder-Decoder (LED)

Longformer by Beltagy *et al.*[52] is a modified Transformer architecture. It addresses the problem of processing lengthy sequences owing to their self-attention operation, *i.e.* the quadratic memory complexity of the transformer. Longformer's attention mechanism combines windowed local-context (sliding-window) self-attention with task-motivated global attention to encode inductive bias regarding the task. As a result, sparse matrix multiplication replaces dense matrix multiplication, resulting in linear complexity concerning input length. The idea is to substitute a sparse attention mechanism that scales linearly with input length, n, for the O(n2) attention mechanism. Masked language modelling (MLM) is used to pre-train Longformer. The Longformer encoder employs the efficient local + global attention pattern instead of complete self-attention to model long sequences for seq2seq learning, and the decoder applies full self-attention to the whole encoded tokens and previously decoded places.[53]

It is pre-trained using BART's exact architecture in terms of the number of layers and hidden size, with the exception that it increases position embedding to 16K tokens (up from 1K tokens in BART) and initializes the new position embedding matrix by cloning BART's 1K position embeddings 16 times. LED comes in two forms: LED-base and LED-large, with 6 and 12 layers in the encoder and decoder stacks, respectively.

The LED uses encoder self-attention, decoder self-attention, and cross-attention in three locations. The length of the input (n) is usually (and frequently considerably) greater than the length of the output (o) (m).

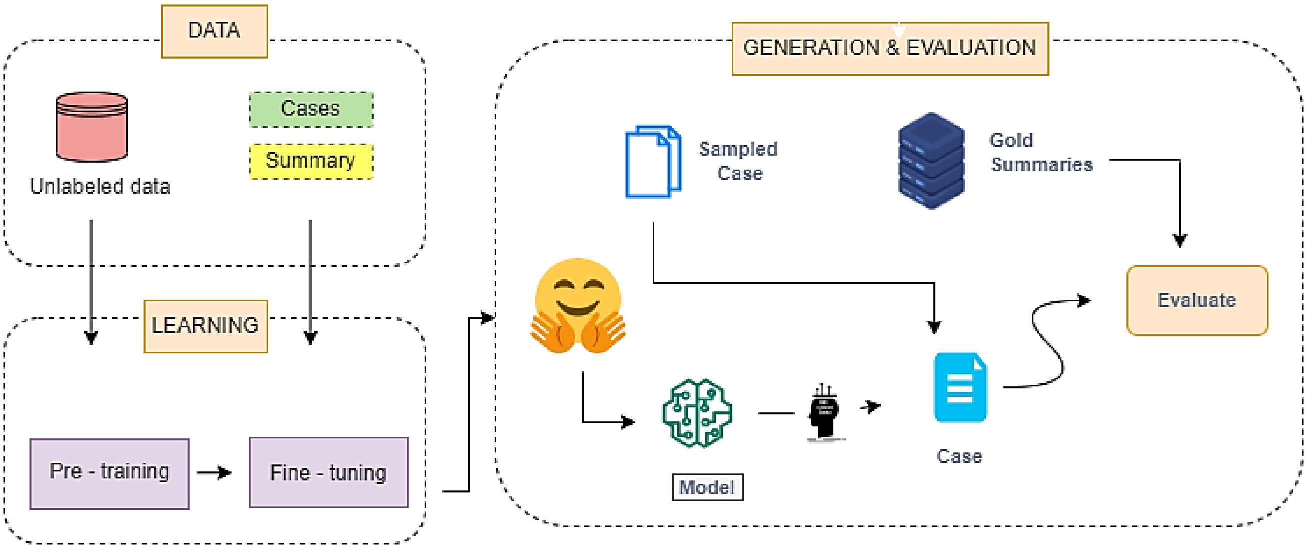

Fig. 8 Our Approach for Abstractive Summarization.

As $n^2 > n \times m > m^2$, the encoder self-attention layer is the critical bottleneck.

4.3 Accuracy measure

We use ROUGE to evaluate the system's performance for the summarizing job; however, evaluating sequence-to-sequence tasks like text summarization is difficult since two equally effective summaries for the same document may exist, each focusing on distinct material and lexically varied.[54]

The most often used summarization assessment measure is ROUGE,[55] which stands for Recall-Oriented Understudy for Gisting Evaluation. It compares a summary generated automatically with reference summaries (typically human-produced). ROUGE comprises various metrics; we utilized ROUGE-1, ROUGE-2 (both instances of ROUGE-N), and ROUGE-L for this thesis.

The number of overlapping words between the reference summary (gold summary, authored by humans) and the system summary (*i.e.* machine-generated summary) is n if we look at individual words. This is a small statistic, but we can use the overlap to calculate the accuracy and recall to achieve a reasonable quantitative number.

Recall (in the context of ROUGE), in the most straightforward words, is the computation of the part of the system summary present in the reference summary. In ROUGE-N, we compute the recall above of n-grams between an output text and a collection of reference text. The formula for computing recall is as follows:

$$Recall = \frac{no\ of\ overlapping\ words}{total\ word\ count\ in\ reference\ summary} \tag{1}$$

This is useful for text summarization but leaves out the opposite side of the issue. A system or machine-generated summary can be quite extensive since it captures all terms in the reference summary. However, many words in the machine-generated summary may need to be revised, making the summary too long. So, accuracy may be used to determine how much of the system overview was useful or required.

$$Precision = \frac{no.of\ overlapping\ words}{total\ word\ count\ in\ system\ summary} \tag{2}$$

When it comes to creating concise summaries, precision becomes a critical factor. As a result, it's advisable to compute both recall and accuracy before reporting the F-Measure, which is the harmonic mean of the two.

- ROUGE-N — It calculates the overlap between unigrams, bigrams, trigrams, and higher-order n-grams.
- ROUGE-L — The most extended word sequence that matches is determined using LCS. There is no need to specify an n-gram length because it automatically includes the longest in-sequence common n-grams.

LCS (largest common sequence) has the benefit of requiring in-sequence matches that represent sentence-level word order rather than merely consecutive matches.

5. Results

A comparative analysis of the dataset uses the techniques and methodologies outlined in Section 4. Multiple domain-independent techniques are used to derive specific key insights, described in this part, as well as the findings of the comparison study, which are displayed in Table 2.

Table 2 contains the results for F-measure and Recall for all three Rouge metrics; for extractive methods, the highest score we achieved is 18.16 using TextRank algorithms, whereas a fine-tuned Longformer-based model achieved 23.18, the highest across all the approaches on the ILC dataset for all the metrics.

6. Discussion

In this work, we presented a new corpus for summarizing Indian legal cases (judgments), and to set a benchmark on the dataset; we experimented with different extractive and abstractive methods such as LSA, TextRank, LED, *etc*., and as we can see from the performance table, transformer-based model *i.e.* led-ILC a variant of Longformer giving the best result compared to extractive methods. LED-ILC archives the rouge2 score of 23.18, which is 27% higher than what was reported by Textrank, a graph-based extractive approach.

For comparison, we used an LED base model, which is pre-trained on standard English corpus, and we got the lowest score; this shows that for legal-specific documents, we need to fine-tune the transformer-based models, and after doing so, we got the highest score this also demonstrates the capability of transfer learning approach.

In abstractive summarization, we utilize the recent advance deep learning methods such as Transformers to reproduce important material in a new way after interpretation and examination of the text; it generates new shorter text that

Table 2. Performance on Test data.

Approach		Rouge-1		Rouge-2		Rouge-L	
		F1	Recall	F1	Recall	F1	Recall
Extractive	Sumbasic	15.69	9.5	6.02	3.5	14.48	9
	LSA	21.20	14.08	7.37	4.6	19.76	13.11
	KLSum	21.40	13.39	10.19	6.15	19.66	12.28
	LexRank	33.21	25.60	16.94	12.65	30.12	23.18
	TextRank	34.63	28.70	18.16	14.62	31.11	25.79
Abstractive	$LED_{(base)}$	4.31	2.28	1.08	0.5	4.11	2.17
	$LED\ ILC_{(our)}$	42.24	40.03	23.18	22.34	39.30	37.25

conveys the most critical information from the original one and produces results are closer to human-like interpretation. Thus, we can get better results using abstractive methods than extractive methods, where we identify important text sections and select the top one as a summary.

7. Conclusion and future work

Very few datasets address the problem of abstractive summarization in the legal domain, and none are specific to Indian legal cases. We propose the new India legal corpus for abstractive summarization and test them against classical and recent powerful deep transformer-based models.

With this paper, we have tried to answer some questions about the need to create a dataset for Indian legal cases for summarization. From the experiment, abstractive summarization methods perform better on long and complex documents than classical extractive methods. Another conclusion that we can draw from this work is when fine-tuned on the ILC dataset, all the model's performance is robust to the difference in language.

This is only the first step in approaching the summarizing of the Indian legal text abstractly; in the future, we can extend this work by creating more datasets with multiple summaries that will allow us to develop the models with more generalization capability.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

Appendix A

Appendix A contains a sample case, gold summary, and machine-generated summary using different methods.

Methods	Text
Case (ID #204)	IN THE HIGH COURT OF DELHI AT NEW DELHI FAO(OS)55 2022 NTPC LTD Appellant Through: Mr Adarsh Tripathi with Mr Ajitesh Garg and Mr Vikram Singh Baid M S TATA PROJECTS LTD Respondent Through: Ms Smita Bhargava with Mr.Tanuj Agarwal and Ms.Pavitra Singh Date of Decision: 03rd March 2022 HON BLE MR. JUSTICE MANMOHAN HON BLE MR. JUSTICE SUDHIR KUMAR JAIN JUDGMENT MANMOHAN J(COMM) 55 2022 & C.M.No.10816 2022 Present appeal has been filed challenging the judgment dated 08 December 2021 passed by learned Single Judge in OMPNo.171 of 2021 whereby the award rendered by learned Arbitral Tribunal was upheld. Learned counsel for Appellant states that the Arbitral Tribunal has grossly erred in not preserving the right of the Appellant to raise its claims for levying liquidated damages subsequently. He states that the learned FAO(OS)55 2022 Single Judge as well as the Arbitral Tribunal failed to consider that the extension given to the Respondent was subject to levy of liquidated damages and the learned Single Judge even failed to note that the Arbitral Tribunal was of the view that the majority of the delay caused in completion of project had been caused by the Respondent. He emphasizes that though no claim was raised by the Appellant in the Arbitral Tribunal qua the liquidated damages yet the Arbitral Tribunal as well as the learned Single Judge that neither party was compensation damages on account of prolongation of the work. A perusal of the paper book reveals that the Respondent claimant had specifically prayed for a declaration that the Respondent claimant was not entitled to levy any liquidated damages for the delay in completion of the project…
Gold Summary	Present appeal has been filed challenging the judgment dated 08 December, 2021 passed by learned Single Judge in OMP (Comm) No.171 of 2021 and was upheld by High Court of Delhi in the case of NTPC LTD vs. M/S TATA PROJECTS LTD (FAO(OS) (COMM) 55/2022) on 03rd March, 2022. Brief facts of the case are that Arbitral Tribunal has grossly erred in not preserving the right of the Appellant to raise its claims for levying liquidated damages subsequently. He states that the learned Single Judge as well as the Arbitral Tribunal failed to consider that the extension given to the Respondent was subject to levy of liquidated damages and the learned Single Judge even failed to note that the Arbitral Tribunal was of the view that the majority of the delay caused in completion of project had been caused by the Respondent. Learned counsel for Appellant emphasizes that though no claim was raised by the Appellant in the Arbitral Tribunal qua the liquidated damages, yet the Arbitral Tribunal as well as the learned Single Judge concluded that neither party was justified in claiming any compensation/damages on account of prolongation of the work. A perusal of the paper book reveals that the Respondent/claimant had specifically prayed for a declaration that the Respondent/claimant was not entitled to levy any liquidated damages for the delay in completion of the project. In

	view of the said specific prayer, this Court is of the opinion that it was incumbent upon the Appellant to prove that the delay was caused by the Respondent/claimant and the Arbitral Tribunal was legally bound to give a finding thereon. In any event, upon a perusal of the award as well as the impugned order, it is apparent that the Appellant had sought to attribute delay in completion of the project on account of the certain action/inaction on the part of the Respondent/claimant. Court is of the view that the intent of the arbitration proceeding, which is an alternative dispute resolution mechanism, is to give a quietus to the dispute. No party can say that even when the Respondent/claimant is seeking liquidated damages on account of delay and had sought a declaration that the Appellant was not entitled to liquidated damages, yet the Appellant would raise its counter-claim before another Arbitral Tribunal. Court finds no ground to interfere with the decision passed by the learned Single Judge. Accordingly, the present appeal along with pending application is dismissed.
KL	He states that the learned FAO(OS)55 2022 Single Judge as well as the Arbitral Tribunal failed to consider that the extension given to the Respondent was subject to levy of liquidated damages and the learned Single Judge even failed to note that the Arbitral Tribunal was of the view that the majority of the delay caused in completion of project had been caused by the Respondent. A perusal of the paper book reveals that the Respondent claimant had specifically prayed for a declaration that the Respondent claimant was not entitled to levy any liquidated damages for the delay in completion of the project. Further this Court is of the view that the intent of the arbitration proceeding which is an alternative dispute resolution mechanism is to give a quietus to the dispute.
SumBasic	He states that the learned FAO(OS)55 2022 Single Judge as well as the Arbitral Tribunal failed to consider that the extension given to the Respondent was subject to levy of liquidated damages and the learned Single Judge even failed to note that the Arbitral Tribunal was of the view that the majority of the delay caused in completion of project had been caused by the Respondent. Accordingly the present appeal along with pending application is dismissed. MARCH 03 2022 MANMOHAN J SUDHIR KUMAR JAIN J FAO(OS)55 2022
LSA	IN THE HIGH COURT OF DELHI AT NEW DELHI FAO(OS)55 2022 NTPC LTD ..... Appellant Through: Mr.Adarsh Tripathi with Mr.Ajitesh Garg and Mr.Vikram Singh Baid M S TATA PROJECTS LTD ..... Respondent Through: Ms.Smita Bhargava with Mr.Tanuj Agarwal and Ms.Pavitra Singh Date of Decision: 03rd March 2022 HON BLE MR. JUSTICE MANMOHAN HON BLE MR. JUSTICE SUDHIR KUMAR JAIN JUDGMENT MANMOHAN J(COMM) 55 2022 & C.M.No.10816 2022 Present appeal has been filed challenging the judgment dated 08 December 2021 passed by learned Single Judge in OMPNo.171 of 2021 whereby the award rendered by learned Arbitral Tribunal was upheld. No party can say that even when the Respondent claimant is seeking liquidated damages on account of delay and had sought a declaration that the Appellant was not entitled to liquidated FAO(OS)55 2022 damages the, Appellant would raise its counterclaim before another Arbitral Tribunal. See: M s. Mangalwar Filling Station v. Indian Oil Corporation Ltd. FAO COMM) 75 2021 dated 07th July 20221.)
LexRank	IN THE HIGH COURT OF DELHI AT NEW DELHI FAO(OS)55 2022 NTPC LTD ..... Appellant Through: Mr.Adarsh Tripathi with Mr.Ajitesh Garg and Mr.Vikram Singh Baid M S TATA PROJECTS LTD ..... Respondent Through: Ms.Smita Bhargava with Mr.Tanuj Agarwal and Ms.Pavitra Singh Date of Decision: 03rd March 2022 HON BLE MR. JUSTICE MANMOHAN HON BLE MR. JUSTICE SUDHIR KUMAR JAIN JUDGMENT MANMOHAN J(COMM) 55 2022 & C.M.No.10816 2022 Present appeal has been filed challenging the judgment dated 08 December 2021 passed by learned Single Judge in OMPNo.171 of 2021 whereby the award rendered by learned Arbitral Tribunal was upheld. He states that the learned FAO(OS)55 2022 Single Judge as well as the Arbitral Tribunal failed to consider that the extension given to the Respondent was subject to levy of liquidated damages and the learned Single Judge even failed to note that the Arbitral Tribunal was of the view that the majority of the delay caused in completion of project had been caused by the Respondent. In view of the said specific prayer, this Court is of the opinion that it was incumbent upon the Appellant to prove that the delay was caused by the Respondent claimant and the Arbitral

	Tribunal was legally bound to give a finding thereon.
TextRank	He states that the learned FAO(OS)55 2022 Single Judge as well as the Arbitral Tribunal failed to consider that the extension given to the Respondent was subject to levy of liquidated damages and the learned Single Judge even failed to note that the Arbitral Tribunal was of the view that the majority of the delay caused in completion of project had been caused by the Respondent. In view of the said specific prayer, this Court is of the opinion that it was incumbent upon the Appellant to prove that the delay was caused by the Respondent claimant and the Arbitral Tribunal was legally bound to give a finding thereon. In any event, upon a perusal of the award as well as the impugned order, it is apparent that the Appellant had sought to attribute delay in completion of the project on account of the certain action inaction on the part of the Respondent claimant
LED ILC	The High Court of Delhi has upheld the judgment passed by learned Single Judge in OMPNo.171 of 2021 whereby the award rendered by learned Arbitral Tribunal was upheld by the Hon'ble Justice Manoj Kumar Jain in the matter of M S TATA PROJECTS LTD. (NTPC) vs. NTPC LTD. The High Court had held that the Arbitration Tribunal has grossly erred in not preserving the right of the Appellant to raise its claims for levying liquidated damages subsequently. The Court observed that the arbitral Tribunal had grossly errred in failing to preserve the right to raise the claims for levy of liquidated damage subsequently. In view of the aforesaid, the Arbitral tribunal has grossly, erred, not preserving its right of Appellants to raise their claims for the liquidation of damages subsequently and the learned Single Court has been legally obliged to render a finding on the said aspect. The court observed that "The Arbitration tribunal grossly errs in not protecting the right" of theAppellant, to raise any claims for paying liquidated/liquidated damages, yet the learned single Judge as well as the learned-single Judge failed to consider that the extension given to the Respondent was subject to levy of the liquidated- damages and the Learned Single Judge even failed to note that the arbitrationral tribunal was of the view that the majority of the delay caused in completion of project had been caused by the Respondant. The bench further observed that though no claim was raised by the Appelant in the Arbitr Tribunal qua the liquidate damages, the learned Senior Judge, who was the arbitrator, had not considered that the Extension given to Respondent, was subjecting to levy the liquidating damages. The learned Single-Judge even failed not to notice that the Appellerant was not entitled to levy any liquidated (refundable) damages. Further, the bench observed that in the present appeal, the Court is of the opinion that it was incumbent upon the Appellant to prove that the delay was caused by a Respondent claimant and the Arbitrator was legally bound to give a finding thereon. In any event, upon a perusal of the award as well and the impugned order, it is apparent that the appellant had sought to attribute delay in completion/ completion of the project on account of the certain action inaction on the part of the Respondenting claimant. Consequently, it was legally obliged to render the finding on said.

References

[1] K. Ježek; Josef Steinberger Automatic Text Summarization (The State-of-the-Art 2007 and New Challenges). In Proceedings of the Znalosti, 2008, 1-12.

[2] N. Munot, S. S. Govilkar, Comparative study of text summarization methods, *International Journal of Computer Applications*, 2014, **102**, 33-37, doi: 10.5120/17870-8810.

[3] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, D. Elizabeth, B. Juan, K. Kochut, Text summarization techniques: a brief survey, *International Journal of Advanced Computer Science and Applications*, 2017, **8**, doi: 10.14569/ijacsa.2017.081052.

[4] H. Lin, V. Ng, Abstractive summarization: A survey of the state of the art, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, **33**, 9815-9822, doi: 10.1609/aaai.v33i01.33019815.

[5] H. P. Luhn, The automatic creation of literature abstracts, *IBM Journal of Research and Development*, 1958, **2**, 159-165, doi: 10.1147/rd.22.0159.

[6] V. Gupta, G. S. Lehal, A survey of text summarization extractive techniques, *Journal of Emerging Technologies in Web Intelligence*, 2010, **2**, 258–268, doi: 10.4304/jetwi.2.3.258-268.

[7] A. Jeet Rawat, S. Ghildiyal, A. K. Dixit, Topic modelling of legal documents using NLP and bidirectional encoder representations from transformers, *Indonesian Journal of Electrical Engineering and Computer Science*, 2022, **28**, 1749, doi: 10.11591/ijeecs.v28.i3.pp1749-1755.

[8] A. Shukla, P. Bhattacharya, S. Poddar, R. Mukherjee, K. Ghosh, P. Goyal, S. Ghosh, Legal case document summarization: extractive and abstractive methods and their evaluation", 2022: arXiv: 2210.07544, doi: 10.48550/arXiv.2210.07544

[9] A. Deroy, K. Ghosh, S. Ghosh, How ready are pre-trained abstractive models and LLMs for legal case judgement

summarization?", 2023: arXiv: 2306.01248. http://arxiv.org/abs/2306.01248.pdf"

[10] A. Kanapala, S. Pal, R. Pamula, Text summarization from legal documents: a survey, *Artificial Intelligence Review*, 2019, **51**, 371-402, doi: 10.1007/s10462-017-9566-2.

[11] F. Galgani, P. Compton, A. Hoffmann, Combining Different Summarization Techniques for Legal Text, Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data, pages, 2021, 115-123.

[12] D. Jain, M. D. Borah, A. Biswas, Summarization of legal documents: where are we now and the way forward, *Computer Science Review*, 2021, **40**, 100388, doi: 10.1016/j.cosrev.2021.100388.

[13] V. Parikh, V. Mathur, P. Mehta, N. Mittal, P. Majumder, LawSum: A weakly supervised approach for Indian Legal Document Summarization", 2021, doi: 10.48550/arXiv.2110.01188.

[14] S. Paul, A. Mandal, P. Goyal, S. Ghosh, Pre-training transformers on Indian legal text, 2022, doi: 10.48550/arXiv.2209.06049.

[15] S. Paul, A. Mandal, P. Goyal, S. Ghosh, Pre-trained language models for the legal domain: a case study on Indian law, Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law. June 19 - 23, 2023, Braga, Portugal. New York: ACM, 2023, 187-196, doi: 10.1145/3594536.3595165.

[16] D. Jain, M. D. Borah, A. Biswas, Summarization of legal documents: where are we now and the way forward, *Computer Science Review*, 2021, **40**, 100388, doi: 10.1016/j.cosrev.2021.100388.

[17] High Courts of India. Wikipedia 2022. Available online: https://en.wikipedia.org/wiki/High_courts_of_India (Accessed on 19 May 2022).

[18] List of District Courts in India. Wikipedia 2022. Available online:
https://en.wikipedia.org/wiki/List_of_district_courts_in_India
(Accessed on 19 May 2022).

[19] M. Saravanan; B. Ravindran; S. Raman Automatic Identification of Rhetorical Roles Using Conditional Random Fields for Legal Document Summarization. In Proceedings of the Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I; 2008.

[20] C. Grover, B. Hachey, C. Korycinski, Summarising legal texts: sentential tense and argumentative rolesProceedings of the HLT-NAACL 03 on Text summarization workshop -. Not Known. Morristown, NJ, USA: Association for Computational Linguistics, 2003, 33–40, doi: 10.3115/1119467.1119472.

[21] M.-Y. Kim, Y. Xu, R. Goebel, Summarization of legal texts with high cohesion and automatic compression rate. New Frontiers in Artificial Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013: 190-204, doi: 10.1007/978-3-642-39931-2_14.

[22] F. Galgani, P. Compton, A. Hoffmann, Citation Based Summarisation of Legal Texts, PRICAI 2012: Trends in Artificial Intelligence: 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, 2012, 40-52, doi: 10.1007/978-3-642-32695-0_6.

[23] A. Joshi, E. Fidalgo, E. Alegre, L. Fernández-Robles, SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders, *Expert Systems With Applications*, 2019, **129**, 200-215, doi: 10.1016/j.eswa.2019.03.045.

[24] R. Srivastava, P. Singh, K. P. S. Rana, V. Kumar, A topic modeled unsupervised approach to single document extractive text summarization, *Knowledge-Based Systems*, 2022, **246**, 108636, doi: 10.1016/j.knosys.2022.108636.

[25] D. Cajueiro, A. G. Nery, I. Tavares, M. K. de Melo, S. A. dos Reis, W. Li, V. R. R. Celestino, A comprehensive review of automatic text summarization techniques: method, data, evaluation and coding, 2023.

[26] Deepa, Anand, Effective deep learning approaches for summarization of legal texts, *Journal of King Saud University - Computer and Information Sciences*, 2022, **34**, 2141-2150, doi: 10.1016/j.jksuci.2019.11.015.

[27] H. Zhang, X. Liu, J. Zhang, Extractive summarization via ChatGPT for faithful summary generation, 2023.

[28] S. Abdel-Salam, A. Rafea, Performance study on extractive text summarization using BERT models, *Information*, 2022, **13**, 67, doi: 10.3390/info13020067.

[29] T. Shi, Y. Keneshloo, N. Ramakrishnan, C. K. Reddy, Neural abstractive text summarization with sequence-to-sequence models, *IMS Transactions on Data Science*, 2021, **2**, 1-37, doi: 10.1145/3419106.

[30] P. Bhattacharya, K. Ghosh, S. Ghosh, A. Pal, P. Mehta, A. Bhattacharya, P. Majumder, FIRE 2019 AILA track: artificial intelligence for legal assistance, Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation. December 12 - 15, 2019, Kolkata, India. New York: ACM, 2019, 4-6, doi: 10.1145/3368567.3368587.

[31] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, M. Sun, Legal Judgment Prediction via Topological LearningProceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium. Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, 3540-3549, doi: 10.18653/v1/d18-1390.

[32] V. Malik, R. Sanjay, S. Nigam, K. Ghosh, S. Guha, A. Bhattacharya, A. Modi, ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation, 2021.

[33] A. Kanapala, S. Pal, R. Pamula, Text summarization from legal documents: a survey, *Artificial Intelligence Review*, 2019, **51**, 371-402, doi: 10.1007/s10462-017-9566-2.

[34] B. Ali, R. More, S. Pawar, G. K. Palshikar, Prior case retrieval using evidence extraction from court judgements. in proceedings of the proceedings of the fifth workshop on automated semantic analysis of information in legal text (ASAIL 2021); ASAIL/LegalAIIA@ICAIL 202: São Paulo, Brazil, June 25 2021.

[35] V. G. D. Gupta, A. Anil, A. S, An ontology driven question answering system for legal documents. 2019 2nd International Conference on Intelligent Computing, Instrumentation and

Control Technologies (ICICICT). July 5-6, 2019, Kannur, India. IEEE, 2020, 947-951, doi: 10.1109/ICICICT46008.2019.8993168.

[36] A Ghosh, P Gupta, R Dutt, K Hiware, A Mandal, K Ghosh, S. Ghosh, Supervised Extraction of Catchphrases from Legal Documents, 2020.

[37] P. Bhattacharya, S. Paul, K. Ghosh, S. Ghosh, A. Wyner, Identification of rhetorical roles of sentences in Indian legal judgments, doi: 10.48550/ARXIV.1911.05405.

[38] V. Eidelman, BillSum: A Corpus for Automatic Summarization of US Legislation, Proceedings of the 2nd Workshop on New Frontiers in Summarization. Hong Kong, China. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, 48–56, doi: 10.18653/v1/d19-5406.

[39] P. Kalamkar, A. Tiwari, A. Agarwal, S. Karn, S. Gupta, V. Raghavan, A. Modi, Corpus for automatic structuring of legal documents, 2022

[40] F. Galgani, A. Hoffmann, Combining Different Summarization Techniques for Legal Text.; Association for Computational Linguistics: Avignon, France, April 2012, 115-123.

[41] L. Manor, J. J. Li, Plain English Summarization of Contracts, Proceedings of the Natural Legal Language Processing Workshop, 2019.

[42] D. De Vargas Feijó, V. P. Moreira, RulingBR: A Summarization Dataset for Legal Texts. In *Computational Processing of the Portuguese Language*; A. Villavicencio, V. Moreira, A. Abad, H. Caseli, P. Gamallo, C. Ramisch, G. Oliveira, G. H. Paetzold, Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2018, 11122, 255-264.

[43] Prime Legal. Available online: https://primelegal.in/blog/ (Accessed on 19 May 2023).

[44] Briefcased. Available online: https://www.briefcased.in/ (Accessed on 26 May 2023).

[45] Lawtimesjournal. Available online: https://lawtimesjournal.in/category/case-summary/ (Accessed on 26 March 2023).

[46] Indiankanoon. Available online: https://indiankanoon.org/ (Accessed on 26 March 2023).

[47] Sumy 2022. (version 0.10.0). Available online: https://github.com/miso-belica/sumy (Accessed on 19 May 2022).

[48] G. Erkan, D. R. Radev, LexRank: graph-based lexical centrality as salience in text summarization, *Journal of Artificial Intelligence Research*, 2004, **22**, 457-479, doi: 10.1613/jair.1523.

[49] R. Mihalcea, P. T. TextRank, Bringing Order into Text. In Proceedings of the Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing; Association for Computational Linguistics: Barcelona, Spain, July 2004, 404–411.

[50] A. Haghighi, L. Vanderwende, Exploring content models for multi-document summarizationProceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '09. May 31-June 5, 2009. Boulder, Colorado. Morristown, NJ, USA: Association for Computational Linguistics, 2009, 362–370, doi: 10.3115/1620754.1620807.

[51] L. Vanderwende, H. Suzuki, C. Brockett, A. Nenkova, Beyond SumBasic: task-focused summarization with sentence simplification and lexical expansion, *Information Processing & Management*, 2007, **43**, 1606-1618, doi: 10.1016/j.ipm.2007.01.023.

[52] J. Steinberger, K. Ježek, Using Latent Semantic Analysis in Text Summarization and Summary Evaluation, In Proceedings of the 7th International Conference ISIM; Ostrava, Czech Republic, April 19, 2004.

[53] Huggingface Transformer 2022. (version 4.19.0). Available online: https://huggingface.co/docs/transformers/index.

[54] I. Beltagy, M. E. Peters, A. Cohan, Longformer: the long-document transformer, 2020. Doi: 10.48550/arXiv.2004.05150.

[55] C. Y. Lin, ROUGE: A Package for Automatic Evaluation of Summaries, In Text Summarization Branches Out, 2004, 74-81.

Publisher's Note: Engineered Science Publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.