Citation-Based Summarization of Landmark Judgments

Purnima Bindal
University of Delhi, India
pbindal@cs.du.ac.in

Vikas Kumar
University of Delhi, India
vikas@cs.du.ac.in

Vasudha Bhatnagar
University of Delhi, India
vbhatnagar@cs.du.ac.in

Parikshet Sirohi
University of Delhi, India
parikshet.sirohi@gmail.com

Ashwini Siwal
University of Delhi, India
asiwal@law.du.ac.in

Abstract

Landmark judgments are of prime importance in the Common Law System because of their exceptional jurisprudence and frequent references in other judgments. In this work, we leverage contextual references available in citing judgments to create an extractive summary of the target judgment. We evaluate the proposed algorithm on two datasets curated from the judgments of Indian Courts and find the results promising.

1 Introduction

In a Common law legal framework, law professionals scrutinize a plethora of legal case documents to comprehend the Court's handling in diverse legal scenarios. These documents customarily span from dozens to hundreds of pages, making them arduous to comprehend. Ergo, condensed summaries are a valuable aid for legal research. Manually crafting case summaries is an intellectually demanding task that requires deep, intense legal knowledge and experience of law experts. Automatic summarization of legal judgments is a practical and potent solution and, therefore, a widely researched problem among NLP researchers. Several earlier studies have provided insights into the challenges associated with summarization of judgments (2004; 2006; 2019; 2021; 2023). Summarizing Indian judgments is tough due to their varied structures, unlike US, UK, Australia and Canada.

Landmark judgments are important court rulings that establish novel legal principles, handle remarkable legal issues, mould the understanding of law, and leave a long-lasting effect on jurisprudence. These judgments not only draw public attention but also gather a large number of citations in later judgments. The citing judgments spotlight the substantive arguments and precedents that lend weight to the ruling in the cited judgment. Each citation in a citing judgment is an informative source for legal points of the prior case, statutes, or laws to support the decision. Summary of a *landmark* judgment highlights the noteworthy attributes of the judgment and provides insight into issues handled within the case, critical legal points, and arguments presented by the lawyers.

Related Work: Legal text summarization has drawn considerable attention, specifically in the Indian context (2006; 2019; 2021; 2021; 2021; 2021; 2022; 2022; 2022; 2022). Bhattacharya et al. compare legal summarization algorithms on Indian Supreme Court judgments dataset[1], while Shukla et al. (2022) comprehensively evaluate extractive and abstractive summarization algorithms through automated and human assessment. Extractive legal-domain specific summarization algorithms in other countries include Farzindar (2004) (Canada), Galgani et al. (2012a); Polsley et al. (2016); Galgani et al. (2012b) (Australia), Liu and Chen (2019) (China), Bhattacharya et al. (2021) (India).

Abstractive Legal Text Summarization algoritrhms include Legal-LED (NSI319, 2021a), Legal-pegasus (NSI319, 2021b), LegalSumm (Feijo and Moreira, 2023). Recently, Paul et al. (2022) developed two transformer-based pretrained language models, InLegalBERT and InCaseLawBERT, through the re-training of LegalBERT (Chalkidis et al., 2021) and CaseLawBERT (Zheng et al., 2021), respectively.

Contributions: We design an unsupervised, extractive algorithm, *CB-JSumm*, to summarize a landmark judgment by leveraging its incoming citations (Sec. 2). We also curate two datasets for citation-based summarization of judgments consisting of 99 (50 + 49) Indian judgments, citing judgments, and gold standard (reference) summaries and assess the quality of the algorithmic summaries(Sec. 3).

[1] https://zenodo.org/record/7152317#.Yz6mJ9JByC0

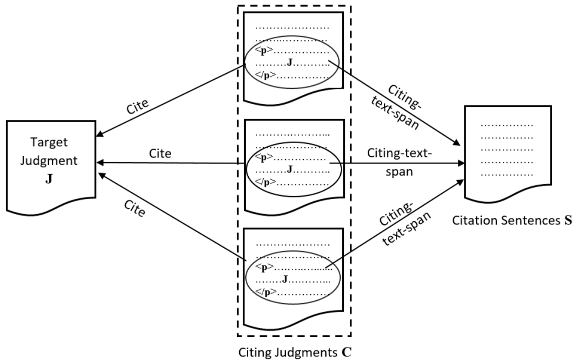

Figure 1: A target landmark judgment $\mathbf{J}$, set $\mathbf{C}$ of citing judgments, and the corpus of extracted citation sentences $\mathbf{S}$

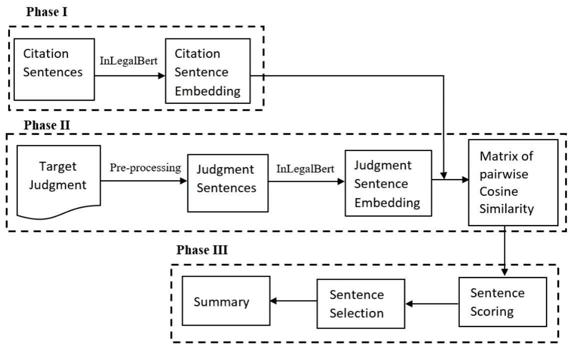

Figure 2: Pipeline for proposed *CB-JSumm* Algorithm

2 Methodology

Given a target landmark judgment $\mathbf{J}$ to be summarized, and $\mathbf{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_t\}$ a set of *citing judgments*, the goal of citation-based legal text summarizer is to present a condensed representation of $\mathbf{J}$ using the *citation* information available in the citing judgments ($\mathbf{C}_i$'s). Judgment $\mathbf{J}$ is tokenized into n sentences $(\mathbf{j_1}, \ldots, \mathbf{j_n})$ and salient sentences are selected for summary $JSumm$.

For citation-based summarization, it is prudent to consider the context in which the judgment is referred. However, extracting context from the citing judgment is a nontrivial task and requires a cautious approach. Since the referring sentence may be inadequate representative of the context in which the target judgment $\mathbf{J}$ is referred, we consider the entire paragraph where the target judgment is cited. This paragraph is referred as the *citing-text-span* and individual sentences in *citing-text-span* are referred as *citation sentences* or simply *citances*.

All *citing-text-spans* are harvested from all citing judgments of $\mathbf{J}$ and tokenized into sentences. The collection of citances from the *citing-text-spans* is denoted by $\mathbf{S}$. Thus, $\mathbf{S} = (\mathbf{s_1}, \ldots, \mathbf{s_m})$ contains all contextual information from all citing judgments contained in $\mathbf{C}$. Fig. 1 clarifies the notation and terminology used throughout the paper.

2.1 Citation Based Judgment Summarization Algorithm

The core idea of the proposed algorithm *CB-JSumm* (*C*itation-*B*ased *J*udgment *Summ*arization) is to leverage contextual information contained in the *citing-text-spans* of the citing judgments for identifying the significant sentences in the target judgment. The algorithm, which has three phases, requires citing judgments of the target judgment ($\mathbf{J}$) to prepare the input corpus of citances ($\mathbf{S}$).

Phase I of the algorithm is the preparatory step to retrieve contextual embeddings of sentences in $\mathbf{S}$. For this purpose, we use InLegalBert, a pre-trained transformer-based language model tailored for the Indian legal domain (Paul et al., 2022). Similarly, we retrieve embeddings for the sentences in $\mathbf{J}$ and compute semantic similarity between the citances and judgment sentences in phase II (Fig. 2). Finally, we identify judgment sentences that are worthy of being included in the summary based on the semantic similarity scores in phase III.

Algorithm 1: *CB-JSumm* Algorithm

Input: set of n judgment sentences $\mathbf{J}$, set of m citation sentences $\mathbf{S}$, desired summary length l
Output: Judgment Summary $JSumm$
1 $S_E \leftarrow$ Embedding($\mathbf{S}$); // Embeddings of m citation sentences
2 $J_E \leftarrow$ Embedding($\mathbf{J}$); // Embeddings of n judgment sentences
3 $\mathcal{S}_{m\times n} \leftarrow$ cosine-sim(S_E, J_E); // Computing similarity score
4 $JSumm \leftarrow$ sentence-scoring($\mathcal{S}$, l); // using Algorithms in Sec. 2.2

Algorithm 1 outlines the proposed *CB-JSumm* algorithm. In Steps 1 and 2, we retrieve contextual embeddings of m citances in $\mathbf{S}$ and n judgment sentences, respectively. In Step 3, we compute the cosine similarity between all pairs of citances and judgment sentences and place them in matrix $\mathcal{S}_{m\times n}$. Thus element s_{pq} of matrix $\mathcal{S}$ denotes the semantic similarity between the p^{th} citance and q^{th} judgment sentence. Next, we employ a sentence scoring method to identify the judgment sentences that are semantically close to the citances and garner the most attention among the citing judgments. The scoring method selects the significant sentences and returns the summary of desired length l. Selected sentences are rearranged according to the judgment's original order to ensure coherence.

The sentence scoring function (Step 4) is the critical component of an unsupervised extractive summarization algorithm and primarily determines the summary quality. Our scoring approach relies on semantic similarity between the contextual information contained in the *citing-text-spans* and the judgment sentences. We describe three scoring heuristics and compare them empirically in Sec. 3.

2.2 Sentence Scoring

In citation-based summarization of judgment, the objective is to identify judgment sentences that closely align with most citances. It is noteworthy that column q in matrix $\mathcal{S}$ reflects the semantic similarity of the q^{th} judgment sentence with all citances. Similarly, row p indicates the similarity of p^{th} citance with all judgment sentences. We propose three approaches described in the subsections below.

2.2.1 CiSumm Sentence Scoring

CiSumm scoring method is a simple and intuitive scoring method that considers all citances equally relevant for creation of summary. The method is based on the intuition that a judgment sentence that exhibits higher overall similarity with all citances deserves to be included in summary. The score of the judgment sentence $\mathbf{j_q}$ is the sum of similarity scores with all citances (i.e. column $\mathcal{S}_{*q}$). Top-scoring judgment sentences are selected as candidates for the summary.

CiSumm scoring scheme is attractive due to its simplicity and efficiency. However, since judgments often have long and complex sentences, longer sentences benefit because of their ability to ensconce richer context. Furthermore, Lengthy judgment sentences sometimes result in unintended semantic similarity with many citances, which may promote them into summary. We examine the effect of length normalization in mitigating the bias due to sentence length in Sec. 3.

2.2.2 Additive Sentence Scoring

Additive sentence scoring method follows the idea that importance of a judgment sentence is proportional to the number of citances with which it bears similarity. Accordingly, we construct a candidate list $\mathcal{C}$ of top-k scoring judgment sentences for each citance. Genuinely important sentences in the judgment are expected to be cited more often and hence may be repeated in the list. Higher number of repetitions indicates that the judgment sentence is semantically similar to more citances, and hence must be important. Similarity scores of the repeated sentences are added to signify their exalted importance. Thus, the additive scoring approach elevates the scores of the candidate sentences in proportion to the references they gather in citing judgments. These sentences are the prime candidates for inclusion in the summary.

Algorithm 2: Additive Sentence Scoring

Input: Similarity Score matrix $\mathcal{S}$, desired summary length l
Output: Judgment Summary $JSumm$
1 For each citation sentence $s_i \in \mathcal{S}$, add top-k scoring judgment sentences along with scores to candidate list $\mathcal{C}$
2 Sum up the scores of each repeated judgment sentence in $\mathcal{C}$
3 Select top-scoring sentences from $\mathcal{C}$ to create summary $JSumm$ of length l;
4 Return $JSumm$

Algorithm 2 outlines the proposed additive scoring method. In Step 1, each row ($\mathcal{S}_{p*}$) of the matrix $\mathcal{S}$ is scanned corresponding to the citance ($\mathbf{s_p}$) and top-k judgment sentences ($\mathbf{j_q}$'s) along with their respective scores (s_{pq}'s) are added to the candidate list $\mathcal{C}$. Step 2 sums up the scores of the repeated judgment sentences, and finally, top-scoring sentences from $\mathcal{C}$ are selected to craft the summary $JSumm$ of desired length l.

2.2.3 Citation Diversity Sentence Scoring

In *C*itation *D*iversity (CD) sentence scoring method, we select judgment sentences while considering the diverse context in which the judgment is cited. This is accomplished by ensuring that each citance (i.e., each row in matrix $\mathcal{S}$) is processed and the corresponding high-scoring judgment sentence is given due consideration.

We select top-k judgment sentences along with their scores for each citance (s_i) and length normalized them. The post-selection length normalization avoids inclusion of shorter judgment sentences, which may not penalized sufficiently if length normalization is done before selection.

For each citation sentence, we arrange judgment sentences in descending order of their length-normalized score. At this juncture, the relevance of the citances is considered by scrutinizing the similarity scores of the top-scoring judgment sentences for all citances. The judgment sentence with the highest score is the one that has the strongest semantic similarity among all citances. The sentence is picked up and added to the candidate set along with the score. Proceeding with a scoring judgment sentence for the next strong citance, the candidate set is created. Finally, top-scoring sentences are selected to construct the summary $JSumm$.

Algorithm 3: Citation Diversity Sentence Scoring

Input: Similarity Score matrix $\mathcal{S}$, desired summary length l
Output: Judgment Summary $JSumm$
1 For each citation sentence $s_i \in \mathcal{S}$, add top-k scoring judgment sentences along with scores to list L_i
2 Normalize similarity score of each judgment sentence in L_i by its length
3 From each L_i, select judgment sentence most similar to s_i and add to candidate set $\mathcal{C}$ if it's not already present
4 Select the sentences from $\mathcal{C}$ in descending order of similarity scores to complete the summary $JSumm$
5 If desired summary length is not completed, revise $\mathcal{C}$ by repeating Step 3 by considering the next most similar judgment sentence from each list $\{L_1, \ldots, L_m\}$ as in Step 3
6 Repeat Steps 4 and 5, until the desired summary length l is achieved
7 Return $JSumm$

Algorithm 3 describes the pseudo-code of the CD scoring method. Step 1 creates the list L_i of k top-scoring judgement sentences along with their corresponding scores for each citance s_i. The scores are revised by normalizing them by the respective sentence length in Step 2. From the list L_i, the judgment sentence that closely matches the citance s_i is included in the candidate set $\mathcal{C}$ if not added yet (Step 3) . This permits efficient comparison of the relevance and diversity of the citances based on their similarities with judgment sentences. Step 4 creates the summary $JSumm$ by choosing the sentences from $\mathcal{C}$ in descending order of scores. If the summary falls shorter than desired summary length l, the next most resembling judgment sentence is added to $\mathcal{C}$ from the lists L_i's (Step 5). Steps 4 and 5 are repeated until the desired summary length is completed (Step 6).

3 Experiment and Results

3.1 Datasets

To assess the performance of *CB-JSumm* algorithm, we curate two datasets[2] tailored for citation-based legal summarization in the Indian context. To the best of our knowledge, no existing dataset leverages judgment citations for summarization of landmark legal judgments.

The first dataset, *IN-Jud-Cit*, consists of fifty landmark judgments from Indian Courts. We meticulously gathered citing judgments from the *Indian Kanoon*[3] website using APIs provided with a free account. For each judgment, the gold-standard summary is obtained from *Casemine*[4] website. Interestingly, these summaries are AI-generated (abstractive) and shorter in length. Hence, we anticipate that the proposed scoring methods will yield a lower ROUGE score with small variation.

The second dataset, *IN-Ext-Cit*, is curated by upgrading the IN-Ext dataset authored by Shukla et al. (2022) and comprising fifty Indian Court judgment-summary pairs. Each judgment in IN-Ext dataset has two associated gold standard summaries written by two different law experts. As before, we obtain citing judgments from the *Indian Kanoon* website for each judgment in IN-Ext dataset. One judgment, for which there was no citation, was discarded, and the remaining 49 judgments were used for experiments. Final score for each evaluation metric is obtained by averaging the score of two reference summaries.

			Judgment Statistics		**Summary Statistics**	
Dataset	J	CJ	$Sent$	W	$Sent$	W
IN-Jud-Cit	50	15	259	8915	20	465
IN-Ext-Cit	49	14	109	3775	47	1375

Table 1: Statistics of two datasets. J: judgments in the dataset, CJ: average number of citing judgments, Sent: median number of sentences, W: median number of words in the judgments

Table 1 summarizes the statistics for judgments, summaries, and citing judgments for both datasets. We report median statistics for words and sentences due to substantial variability in judgment lengths within the datasets. It is observed that judgments in *IN-Jud-Cit* dataset are lengthier than those in *IN-Ext-Cit*. The reference summaries in *IN-Jud-Cit* dataset are approximately 5% of the original judgment length, whereas *IN-Ext-Cit* summaries are approximately 36% of the judgment length.

3.2 Competing Methods and Evaluation Metrics

We assess *CB-JSumm's* performance for three sentence scoring methods (with and without sentence length normalization), with four competing legal-domain algorithms: CaseSummarizer(Polsley et al., 2016), MMR(Shukla et al., 2022), Legal-pegasus(NSI319, 2021b) and Legal-LED(NSI319, 2021a), obtained from GitHub repository[5].

We report macro-averaged ROUGE F-scores (ROUGE-1, ROUGE-2 and ROUGE-L). We augment our investigation by adopting the semantic-based assessment metric introduced by Steinberger and Ježek (2009).

[2] https://github.com/PurnimaBindal/LegalTextSummarization
[3] https://indiankanoon.org/
[4] https://www.casemine.com/
[5] https://github.com/Law-AI/summarization

Algorithm	Scoring Methods	*IN-Jud-Cit* Dataset			*IN-Ext-Cit* Dataset		
		ROUGE-F Scores					
		Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
CB-JSumm	CiSumm (§ 2.2.1)	$\mathbf{52.13 \pm 6.46}$	19.10 ± 6.44	54.48 ± 6.39	64.60 ± 5.91	38.62 ± 9.34	66.74 ± 5.68
	CiSumm-LN (§ 2.2.1)	49.40 ± 7.46	18.06 ± 7.63	53.79 ± 6.88	65.05 ± 6.13	40.22 ± 9.91	67.95 ± 5.66
	Additive (§2.2.2)	50.83 ± 5.99	17.93 ± 5.14	53.39 ± 5.69	64.15 ± 6.38	38.00 ± 9.47	66.37 ± 6.07
	Additive-LN (§2.2.2)	50.27 ± 7.35	18.07 ± 7.19	53.30 ± 6.90	64.85 ± 5.85	39.17 ± 9.09	67.25 ± 5.44
	CD (§2.2.3)	51.65 ± 6.20	$\mathbf{19.90 \pm 7.12}$	$\mathbf{55.61 \pm 5.71}$	$\mathbf{65.49 \pm 5.98}$	$\mathbf{40.29 \pm 9.38}$	$\mathbf{68.04 \pm 5.70}$
Competing	CaseSummarizer	40.17 ± 7.43	10.16 ± 6.25	44.11 ± 6.86	59.44 ± 6.28	31.64 ± 8.58	61.94 ± 6.00
	MMR	51.48 ± 8.93	19.79 ± 8.58	54.72 ± 8.44	57.80 ± 6.30	27.97 ± 8.44	60.38 ± 6.04
	Legal-pegasus	47.76 ± 13.63	18.28 ± 8.79	50.41 ± 13.21	61.97 ± 5.84	32.77 ± 8.12	64.20 ± 5.61
	Legal-LED	37.89 ± 6.01	10.15 ± 3.47	42.36 ± 5.50	49.03 ± 5.42	22.96 ± 5.89	52.55 ± 5.05

Table 2: Macro-averaged ROUGE F-scores along with std. deviation for the two datasets. LN: Length-normalization.

3.3 Experimental Results

We report experimental results for two datasets separately, as macro-averaged values along with the standard deviations.

Results for IN-Jud-Cit Dataset: Table 2 exhibits *CB-JSumm's* superior performance over four competing methods. All scoring methods surpass CaseSummarizer, which is corpus-dependent and uses *tf-idf* for sentence scoring. ROUGE scores for CD sentence scoring method are competitive with MMR scores, but other scoring methods show marginal performance decline. MMR algorithm also employs *tf-idf* for sentence scoring and achieves diversity by repetitively scoring judgment sentences, which slows down the algorithm. Legal-pegasus and Legal-LED's degraded performance aligns with Shukla et al. (2022) results.

As evident from Table 3, the citation-diversity scoring method outperforms other scoring methods and competing algorithms. MMR and Legal-Pegasus perform better than additive-scoring methods, while the performances of CaseSummarizer and Legal-LED leave much to be desired.

Algorithm	Scoring Methods	*IN-Jud-Cit*	*IN-Ext-Cit*
CB-JSumm	CiSumm (§ 2.2.1)	0.75 ± 0.12	$\mathbf{0.93 \pm 0.05}$
	CiSumm-LN (§ 2.2.1)	0.76 ± 0.11	0.92 ± 0.04
	Additive (§ 2.2.2)	0.74 ± 0.11	$\mathbf{0.93 \pm 0.04}$
	Additive-LN (§ 2.2.2)	0.74 ± 0.12	$\mathbf{0.93 \pm 0.04}$
	CD (§ 2.2.3)	$\mathbf{0.78 \pm 0.09}$	$\mathbf{0.93 \pm 0.05}$
Competing	CaseSummarizer	0.44 ± 0.18	0.90 ± 0.08
	MMR	0.76 ± 0.16	0.92 ± 0.05
	Legal-pegasus	0.76 ± 0.19	0.92 ± 0.05
	Legal-LED	0.60 ± 0.13	0.65 ± 0.13

Table 3: Macro-averaged semantic similarity scores with standard deviation between system and reference summary for both datasets. LN: Length-normalization.

Results for IN-Ext-Cit Dataset: ROUGE scores are comparatively higher for this dataset owing to longer summaries (Table 2). *CB-JSumm* consistently outperforms competing algorithms with a bigger margin due to similar number of citing judgments and considerably shorter judgment length than *IN-Jud-Cit* dataset (Table 1). This not only vindicates the importance of citations for summarization of landmark judgments, it also explains the narrow winning margin for *IN-Jud-Cit* dataset, where the judgments are much longer but adequate citations are not available for summarization. Further, CD scoring method slightly outperforms other scoring methods across all ROUGE variations.

Semantic similarity scores for this dataset are also higher due to lengthy summaries for shorter judgments. As evident in Table 3, *CB-JSumm* algorithm performs better than other competing methods. Legal-LED and CaseSummarizer under perform, while MMR and Legal-pegasus slightly lag behind the proposed algorithm.

As stated before, *IN-Jud-Cit* dataset reference summaries are short and abstractive, limit ROUGE score diversity among the proposed sentence scoring methods. For *In-Ext-Cit* dataset, CD scoring performs best among all, as the reference summary contains sentences of the original judgment. Higher scores of this method indicate that post-selection length-normalization excludes very short judgment sentences favored by pre-selection normalization.

4 Conclusion

We propose *CB-JSumm*, an extractive and unsupervised algorithm to summarize landmark judgments leveraging contextual information from citing judgments. We evaluate proposed algorithm using two curated dataset and observe encouraging results.

Acknowledgements

We thank authors Shukla et al. (2022) for sharing IN-Ext dataset. We also extend our sincere appreciation to Vineet Kumar and Irfan Sheikh for their dedicated efforts in developing the dataset pipeline.

References

Deepa Anand and Rupali Wagh. 2022. Effective deep learning approaches for summarization of legal texts. *Journal of King Saud University-Computer and Information Sciences*, 34(5):2141–2150.

Paheli Bhattacharya, Kaustubh Hiware, Subham Rajgaria, Nilay Pochhi, Kripabandhu Ghosh, and Saptarshi Ghosh. 2019. A comparative study of summarization algorithms applied to legal case judgments. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41*, pages 413–428. Springer.

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

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2021. Lexglue: A benchmark dataset for legal language understanding in english. *arXiv preprint arXiv:2110.00976*.

Atefeh Farzindar. 2004. Atefeh farzindar and guy lapalme,'letsum, an automatic legal text summarizing system in t. gordon (ed.), legal knowledge and information systems. jurix 2004: The seventeenth annual conference. amsterdam: Ios press, 2004, pp. 11-18. In *Legal knowledge and information systems: JURIX 2004, the seventeenth annual conference*, volume 120, page 11. IOS Press.

Diego de Vargas Feijo and Viviane P Moreira. 2023. Improving abstractive summarization of legal rulings through textual entailment. *Artificial intelligence and law*, 31(1):91–113.

Shaz Furniturewala, Racchit Jain, Vijay Kumari, and Yashvardhan Sharma. 2021. Legal text classification and summarization using transformers and joint text features.

Filippo Galgani, Paul Compton, and Achim Hoffmann. 2012a. Citation based summarisation of legal texts. In *PRICAI 2012: Trends in Artificial Intelligence: 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, September 3-7, 2012. Proceedings 12*, pages 40–52. Springer.

Filippo Galgani, Paul Compton, and Achim Hoffmann. 2012b. Combining different summarization techniques for legal text. In *Proceedings of the workshop on innovative hybrid approaches to the processing of textual data*, pages 115–123.

Satyajit Ghosh, Mousumi Dutta, and Tanaya Das. 2022. Indian legal text summarization: A text normalization-based approach. In *2022 IEEE 19th India Council International Conference (INDICON)*, pages 1–4. IEEE.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. 2021. Summarization of indian legal judgement documents via ensembling of contextual embedding based mlp models.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022. Named entity recognition in indian court judgments. *arXiv preprint arXiv:2211.03442*.

Chao-Lin Liu and Kuan-Chun Chen. 2019. Extracting the gist of chinese judgments of the supreme court. In *proceedings of the seventeenth international conference on artificial intelligence and law*, pages 73–82.

NSI319. 2021a. Legal language model (led) base 16384. `https://huggingface.co/nsi319/legal-led-base-16384`.

NSI319. 2021b. Legal pegasus. `https://huggingface.co/nsi319/legal-pegasus`.

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

Shounak Paul, Arpan Mandal, Pawan Goyal, and Saptarshi Ghosh. 2022. Pre-training transformers on indian legal text. *arXiv preprint arXiv:2209.06049*.

Seth Polsley, Pooja Jhunjhunwala, and Ruihong Huang. 2016. Casesummarizer: A system for automated summarization of legal texts. In *Proceedings of COLING 2016, the 26th international conference on Computational Linguistics: System Demonstrations*, pages 258–262.

Murali Saravanan, Balaraman Ravindran, and Shivani Raman. 2006. Improving legal document summarization using graphical models. *Frontiers in Artificial Intelligence and Applications*, 152:51.

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 *The 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*.

Josef Steinberger and Karel Ježek. 2009. Evaluation measures for text summarization. *Computing and Informatics*, 28(2):251–275.

Lucia Zheng, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 159–168.