A Comparative Study of Summarization Algorithms Applied to Legal Case Judgments

Chapter *in* Lecture Notes in Computer Science · April 2019

DOI: 10.1007/978-3-030-15712-8_27

CITATIONS
106

READS
4,443

6 authors, including:

Kaustubh Hiware
Indian Institute of Technology Kharagpur
11 PUBLICATIONS **753** CITATIONS
SEE PROFILE

Subham Rajgaria
Indian Institute of Technology Kharagpur
6 PUBLICATIONS **289** CITATIONS
SEE PROFILE

Kripabandhu Ghosh
Indian Statistical Institute
94 PUBLICATIONS **1,564** CITATIONS
SEE PROFILE

Saptarshi Ghosh
Indian Institute of Technology Kharagpur, India
212 PUBLICATIONS **4,609** CITATIONS
SEE PROFILE

A Comparative Study of Summarization Algorithms Applied to Legal Case Judgments

Paheli Bhattacharya[1(✉)], Kaustubh Hiware[1], Subham Rajgaria[1], Nilay Pochhi[1], Kripabandhu Ghosh[2], and Saptarshi Ghosh[1]

[1] Indian Institute of Technology Kharagpur, Kharagpur, India
pahelibhattacharya@gmail.com
[2] Indian Institute of Technology Kanpur, Kanpur, India

Abstract. Summarization of legal case judgments is an important problem because the huge length and complexity of such documents make them difficult to read as a whole. Many summarization algorithms have been proposed till date, both for general text documents and a few specifically targeted to summarizing legal documents of various countries. However, to our knowledge, there has not been any systematic comparison of the performances of different algorithms in summarizing legal case documents. In this paper, we perform the first such systematic comparison of summarization algorithms applied to legal judgments. We experiment on a large set of Indian Supreme Court judgments, and a large variety of summarization algorithms including both unsupervised and supervised ones. We assess how well domain-independent summarization approaches perform on legal case judgments, and how approaches specifically designed for legal case documents of other countries (e.g., Canada, Australia) generalize to Indian Supreme Court documents. Apart from quantitatively evaluating summaries by comparing with gold standard summaries, we also give important qualitative insights on the performance of different algorithms from the perspective of a law expert.

Keywords: Summarization · Legal case judgment · Supervised · Unsupervised

1 Introduction

In countries following the *Common Law* system (e.g., UK, USA, Canada, Australia, India), there are two primary sources of law – *Statutes* (established laws) and *Precedents* (prior cases). Precedents help a lawyer understand how the Court has dealt with similar scenarios in the past, and prepare the legal reasoning

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-15712-8_27) contains supplementary material, which is available to authorized users.

L. Azzopardi et al. (Eds.): ECIR 2019, LNCS 11437, pp. 413–428, 2019.
https://doi.org/10.1007/978-3-030-15712-8_27

accordingly. Hence lawyers have to go through hundreds of prior cases. These cases are available as law reports/case judgments which are essentially long[1] and free-flowing with dense legal text.

Table 1. Performances of summarization approaches on legal documents of various countries, as reported in the corresponding papers.

Method	Corpus/#Documents	ROUGE-1	ROUGE-2	ROUGE-L
LetSum [15]	Federal Court of Canada/3500	0.575	0.313	0.451
CaseSummarizer [38]	Federal Court of Australia/5	0.194	0.114	0.061
Graphical Model [40]	Kerala High Court/200	0.6	0.386	–

This makes reading and comprehending the full text of a case a difficult task, even for a legal expert. In scenarios like this, summaries of the case judgments prove to be beneficial.

All popular legal retrieval systems provide summaries of case judgments. Due to the complexity of case documents, they are mostly manually summarized by legal experts. For instance, the popular Westlaw India legal system employs legal attorneys to summarize Indian legal documents [6]. Employing experts to write the summaries incurs high cost. Hence, with the advancement of the Web and large amounts of unstructured legal documents being made available everyday, there is an increasing need for *automated legal text summarization* that can work in such dynamic environments.

In this work, we explore the task of automatic summarization of Indian legal case judgments, specifically of the Supreme Court of India. It can be noted that several prior works have developed text summarization techniques on various legal text, e.g., Canadian case judgments [14,15], UK case judgments (House of Lords Judgments) [21,25], legal judgments from Indian High Courts [40], judgments from the Federal Court of Australia [38], and so on. However, *there has not been any systematic investigation of whether methodologies developed for legal text of one country, generalize well to legal text of another country.* Different countries have their own formats for law reports, and legal terminologies vary widely between different countries. Hence, a summarization approach targeted to case judgments of one country may not generalize well to case judgments from other countries.

Table 1 shows the performance of case document summarization algorithms developed for documents of different countries/courts. The performance measures (ROUGE scores) stated by the corresponding prior works are also mentioned. The datasets used in the prior works differ significantly on the nature of legal documents, size of corpus, and so on. No systematic comparisons have been performed regarding the generalizability of these methods developed for

[1] The average length of an Indian Supreme Court judgment is as high as 4,500 words. Important 'landmark' cases often span hundreds of pages, e.g. https://indiankanoon.org/doc/257876/.

documents of one country/court to documents of other countries/courts, and it is not clear which of the methods would perform well in summarizing documents for a different country/court. To bridge this gap, in this work, we apply all these (and many other) methods in a common setting, namely summarizing Indian Supreme Court case judgments. Note that Australian [1], Canadian [2] and UK [5] law reports have *section headings* and follow a certain structure, whereas Indian case law reports do not usually contain any such information and are highly unstructured, making the summarization task more challenging[2].

Additionally, there are a large number of domain-independent summarization algorithms [23,36] – including classical unsupervised algorithms as well as recent supervised neural algorithms – which can potentially be used for summarizing legal case judgements. Again, there has not been any systematic investigation of how well domain-independent summarization algorithms perform on legal text. In this scenario, we in this paper make the following contributions:

- **Generalizability and Classical Reproducibility:** We assess the performance of several domain-independent text summarization methods (both traditional unsupervised ones and recent supervised neural models) on legal case judgments. We also assess how well summarization algorithms developed for documents of one country generalize to documents of another country. Specifically, we reproduce three existing extractive text summarization algorithms specifically designed for legal texts, two for documents of other countries (Canada and Australia) and one for Indian case documents of another court. The implementations of the algorithms explored in this paper are available at https://github.com/Law-AI/summarization.
- **Comparative evaluation:** We perform an extensive evaluation of the performance of different summarization algorithms on Indian Supreme Court case judgments. The evaluation is both quantitative (in terms of comparing with gold standard summaries using ROUGE scores) and qualitative (gathering opinion from legal experts). We show that there is no one best performing summarization algorithm for legal case judgments. While one method performs quantitatively better, another can generate a qualitatively better summary.

To the best of our knowledge, this is the first work that performs a systematic comparison of the performance and generalizability of legal document summarization methods in a common setting.

2 State-of-the Art on Legal Document Summarization

We have classified the prior works into two broad categories – (i) summarization algorithms specifically for legal text, and (ii) domain-independent summarization approaches. We describe these two types of prior works in this section.

[2] https://timesofindia.indiatimes.com/india/when-even-judges-cant-understand-judgments/articleshow/58690771.cms.

2.1 Summarization Algorithms Specifically for Legal Documents

Text summarization approaches have been applied to legal texts of many countries. The survey paper [28] highlights research in this field. Existing methods for summarization of legal text can be broadly classified into (i) unsupervised, (ii) supervised and (iii) citation based approaches.

Unsupervised Approaches: These methods use linguistic and statistical signals from the text to identify important sentences for summarization. Initial attempts at summarizing legal cases was by [19,32]. A recent work on unsupervised legal text summarization infusing additional domain knowledge is *CaseSummarizer* [38]. Since we reproduce this method, details can be found in Sect. 4.2.

Supervised Approaches: Supervised approaches for legal text summarization perform a type of template-filling task. Here, the templates are the rhetorical roles (e.g., facts of the case, background, precedent and statutes, arguments, verdict of the Court, etc.). Each of these slots are filled with sentences, ranked in order of their importance. The *LetSum* project [14,15] and a method using *Graphical models* [40] have been applied for legal case document summarization. Since we reproduce these methods, details can be found in Sect. 4.2. The *Sum* project [21,22,24,25] uses several linguistic features and various machine learning techniques to classify a sentence into one of the rhetorical role labels. For summary generation, they select sentences located at the periphery of each rhetorical category.

Citation based approaches leverage other documents to summarize a target document, e.g. [17]. For a target document, they use the catchphrases of the documents cited by the target document (citphrases) and the citation sentences of documents that cite the target document (citances). Another work by the same authors [18] propose to combine these with a Knowledge Base using Ripple Down Rules that suggest different parameters based on which a sentence is to be chosen for summary.

2.2 Domain-Independent Text Summarization Algorithms

Many domain-independent text summarization algorithms have been proposed, as covered in several survey papers [7,11,23,36].

Classical Extractive Text Summarization Methods: There is a wide variety of methods, of which we describe a few. One of the earliest approaches for text summarization is *Luhn's method* [29]. There are graph-based [13,31] and matrix-based approaches [20] for summary generation. The data reconstruction approach *(DSDR)* [26] generates the summary by extracting those sentences that are more probable in reconstructing the original document. It selects sentences that minimize the reconstruction error.

Neural Network Based Summarization Algorithms: In recent years, Deep Learning has been applied to text summarization; see [12] for a survey. Supervised deep neural architectures have been proved to be extremely beneficial

for generating **abstractive summaries** [10,34,39,41]. Reinforcement learning have also been applied to abstractive summarization [35,37]. Neural models have also been used for **extractive summarization** [8,9,27,33,43]. An unsupervised extractive text summarization algorithm using Restricted Boltzmann Machines (RBM) was developed in [42].

3 Data and Experimental Setup

Dataset Details: We collected 17,347 legal case documents of the Supreme Court of India from the years 1990–2018 from the website of Westlaw India (http://www.westlawindia.com). For each case judgment, Westlaw provides the full text judgment and a summary. Summaries are written by legal attorneys employed by Westlaw [6]. We use these summaries as gold standard summaries for evaluation of algorithmically generated summaries. Each document has 4,533 words and 116 sentences on average.

Training Data: We use the chronologically earlier 10,000 documents as the training set. For instance, the neural abstractive text summarization algorithm explored in this work, is trained over these 10,000 documents and their gold standard summaries (details in Sect. 4.1).

Test Data: The remaining 7,347 case documents (chronologically later ones) are used as the test set. We generate summaries and perform all quantitative evaluations on the test set. Note that we split the train-test datasets based on chronological ordering of the cases because, in practice, models trained over past cases will be applied to future cases.

Summary Length: Some algorithms require the desired length of the summary to be given as an input. For each document (in the test set), we fix the desired length of the summary to 34% of the number of words in the full text judgment of the document. This number was chosen based on the average ratio of the number of words in the gold standard WestLaw summaries and the original documents, over the entire collection.

4 Applying Summarization Algorithms to Indian Legal Case Judgments

This section describes the application of several text summarization algorithms to Indian Supreme Court legal case judgments.

4.1 Domain-Independent Text Summarization Algorithms

Traditional Unsupervised Extractive Methods: We used the publicly available implementations of LSA, LexRank (both available at [4]), Frequency Summarizer [16] and the data reconstruction method DSDR [3] for summarizing the legal documents.

Neural Network Based Extractive Summarization Method: Most of the supervised neural extractive text summarization methods require *sentence level annotations* regarding the suitability of the inclusion of the sentence in the summarization. For example, [9] uses a 0/1/2 annotation for each sentence denoting whether it should not be, may be, or should be included in the summary. Such sentence-level annotation for legal case judgments can only be done by legal experts, and the cost would be prohibitively expensive due to the large length of case judgments (116 sentences on average). Hence, we use the *unsupervised* model based on Restricted Boltzmann Machines [42] whose implementation is publicly available. We use the default parameter settings – a single hidden layer with 9 perceptrons each having learning rate of 0.1. We increased the training epochs to 25.

Neural Network Based Abstractive Summarization Method: We use the pointer generator approach for abstractive text summarization [41] that uses deep learning architectures (implementation publicly available). We trained on 10,000 documents for 18,729 epochs (over five days). The learning rate was initialized to 0.15 and it fell to 0.00001 with training. The number of decode steps are increased from 100 to 150 to incorporate more decoding words. The size of the vocabulary was increased from 50,000 to 2,00,000 since legal case documents are large. All other parameters were set to default.

4.2 Summarization Algorithms Specifically for Legal Documents

From the family of *unsupervised* algorithms (as described in Sect. 2), we reproduce the model of CaseSummarizer [38] as it is a more recent approach. The methods leveraging citations employ multiple citing and cited documents to summarize a particular document. Since all the other methods aim at summarizing a particular document using linguistic signals from that document alone (and does not use other documents), we do not consider these methods in this paper, for a fair performance evaluation. From the family of *supervised* algorithms, we reproduce the Graphical model based approach [40], because the authors have experimented on cases from Kerala High Court (Kerala is an Indian state) which would intuitively be similar to those of the Indian Supreme Court (over which we are experimenting). We also reproduce the LetSum model [15]; we choose this method over the SUM model as they provide more and understandable technical content.

Generalization Across Legal Documents of Various Countries: Note that CaseSummarizer [38] was developed for Australian legal documents and LetSum [15] was developed for Canadian legal documents. We want to understand how these algorithms generalize to documents from another country (India) and what modifications are necessary to adopt the methods.

A Challenge in Reproducing Supervised Legal Summarization Algorithms: As stated in Sect. 3, since supervised algorithms perform a slot-filling task, it is necessary to decide the rhetorical categories of sentences in a

case judgment, before manual annotation. Different prior works on legal text use different rhetorical categories, as shown in Table 2. The FIRE Legal Track [30] developed a scheme of rhetorical categories for Indian Supreme Court cases (Table 2 last column), and we chose to use this annotation scheme in reproducing the methods. We also noted that, although different works use different rhetorical schemes, there is a semantic mapping between the various schemes. We use the mapping from Table 3 (developed in discussion with legal experts) while reproducing the prior works GraphicalModel [40] and LetSum [15].

Table 2. Rhetorical categories of sentences in legal case judgments, as identified in different prior works

GraphicalModel [40]	LetSum [15]	FIRE Legal Track [30]
Identifying facts, Establishing facts, Arguing, History, Arguments, Ratio, Final Decision	Introduction Context Juridical Analysis Conclusion	Fact, Issue, Argument, Ruling by lower court, Statute, Precedent, Other general standards, Ruling by the present court

We now describe how we reproduced the three summarization methods specifically for legal documents. We will make the implementations of these methods publicly available upon acceptance of the paper.

4.2.1 Unsupervised Approach: CaseSummarizer [38]

Basic Technique: Standard preprocessing techniques are done using the NLTK library. Each word is then weighted using a TF-IDF score. For each sentence, the TF-IDF values of its constituent words are summed up and normalized over the sentence length. This score is called w_{old}. A new score, w_{new}, is computed for the sentence using $w_{new} = w_{old} + \sigma\,(0.2d + 0.3e + 1.5s)$ where d is the number of 'dates' present in the sentence, e is the number of named entity mentions, s is a boolean indicating the start of the section (sentences at the start of a section are given more weightage), and σ is the standard deviation among the sentence scores.

Challenges in Reproducibility: Unlike Australian case judgments, Indian case judgments are much less structured, and do not contain section/paragraph headings. As an alternative estimate of the importance of a sentence, we used a count of the number of legal terms (identified by a legal dictionary) present in the sentence. The importance of 'dates' was not clear, and Indian case judgments have very few dates. Rather, Indian case judgments refer to Sections of particular Acts in the Indian legal system, e.g., 'section 302 of the Indian Penal Code'. Hence, for the parameter d in the formulation, we included both dates and section numbers.

Table 3. Mappings between rhetorical categories in different works

Mapping to GraphicalModel	
FIRE Legal Track [30]	**GraphicalModel [40]**
Facts	Identifying facts
Issue	Establishing facts
Precedent	Arguing
Ruling by lower court	History
Arguments+Other general standards	Arguments
Statute	Ratio
Ruling by present court	Final Decision

Mapping to LetSum	
FIRE Legal Track [30]	**LetSum [15]**
Facts + Issue	Introduction
Arguments + Ruling by lower court	Context
Statute + Precedent	Juridical Analysis
Other general standards+ Ruling by present court	Conclusion

Challenges in Applying Standard NLP Tools to Legal Texts: There is another set of challenges in applying standard NLP tools to legal texts. For instance, the authors of [38] did not clearly mention how they identified the 'entities' in the texts. So, we used the popular Stanford NER Tagger[3] for identifying named entities. We found that the tool gives many false positives. For instance, the phrase 'Life Insurance Corporation India' actually represents a single organization. But Stanford NER identifies 'Life : PERSON' , 'Insurance Corporation : ORGANIZATION', 'India : LOCATION'. Again the phrase 'Pension Rules' is identified as a PERSON. Also, we find that using the popular Python NLTK library[4] for tokenizing a legal document poses many difficulties. For instance, in legal documents, a lot of abbreviations are present. NLTK attempts to use 'fullstops' as boundaries, resulting in many incorrect parses. An example: the phrase *"issued u/s. 1(3) of the Act"* is tokenized to *'issued'*, *'u/s'*, '.' , *'1'*, *'('*, *'3'*, *')'*, *'of'*, *'the'*, *'Act'*.

4.2.2 Supervised Approach: LetSum [14,15]

Basic Technique: LetSum divides the text structure into five themes as mentioned in Table 2. The summary is built in four phases: (i) thematic segmentation, (ii) filtering of less important textual units including case citations, (iii) selection of candidate units, and (iv) production of the summary. Sentences are assigned a theme based on the presence of hand-engineered linguistic markers. Citation units are filtered out, which are identified by presence of numbers, certain prepositions and markers like colons, quotations, etc. A list of best candidate units for each structural level of the summary is selected, based on heuristic functions, locational features, and TF-IDF of the sentence. The final summary is produced by concatenating textual units with some manual grammatical modifications. The Introduction forms 10% of the size of summary, the context is 24%, Juridical analysis and Conclusion segments are 60% and 6% of the summary respectively.

Note that, the Production module (the last phase, which deals with manually making grammatical modifications to the selected words) was mentioned as being

[3] https://nlp.stanford.edu/software/CRF-NER.shtml.
[4] https://www.nltk.org/.

implemented in the papers [14,15], and no related future work from the authors could be found. Hence this step had to be omitted.

Challenges in Reproducibility: Indian case documents do not have a fixed structure and lack section headings, unlike Canadian documents. Also, citations to statutes are important for summaries of Indian legal case judgments; thus, we do *not* carry out the phase where citations are filtered out.

Since the writing style for judicial texts in both countries are widely different, *the linguistic markers identified for Canadian legal texts could not identify themes for sentences in Indian legal documents*. Thus, we extract cue phrases as follows. We randomly selected 25 documents from the training set (described in Sect. 3) for manual annotation by legal experts. Based on these 25 annotated documents, we rank the most frequent n-grams in a theme which are minimally present in other themes. This part heavily relies on manual annotation (which is expensive in legal domain), which we have tried to automate to a large extent in our reproduction.

Also, according to the LetSum algorithm, each sentence of the document is assigned to a theme. Within each theme, sentences are ranked based on a heuristic function. However, the *heuristic function was not specified in* [14,15], in the absence of which we ranked sentences based on their TF-IDF scores. For each theme, the maximum length in the summary is known. Hence an adequate number of textual units are correspondingly chosen. Additionally the problems of using NLTK (as stated above) are encountered here as well.

4.2.3 Supervised Approach: GraphicalModel [40]

Basic Technique: The authors identified the rhetorical roles of a sentence using Conditional Random Fields. The features identified for each sentence are presence of indicator/cue phrases, position of particular words in the sentence (beginning/end of the sentence, index) and layout features such as position of sentence in the document, capitalization, presence of digits and Part-of-Speech tags. The term distribution model (the k-mixture model) is used to assign probabilistic weights. Sentence weights are computed by summing the term probability values obtained by the model. Sentences are subsequently re-ranked twice, once based on their weights and again based on their evolved roles during CRF implementation, to generate the final summary.

Challenges in Reproducibility: Since the original paper focused on cases related to 'rent control' and the annotations available with us were not from rent-specific cases, *the cue phrases mentioned in* [40] *did not perform well in our dataset*. Hence we automate the process of identification of cue phrases as follows. For all the annotated documents, sentences were separated based on their annotations (by legal experts). Identification of cue phrases for each category was achieved by computing n-grams, along with their frequency in the specific category. An n-gram was chosen as a cue phrase for a particular role label, if its frequency across all the other categories was lower.

Default values of parameters were used for the CRF (implemented using the Python library 'pycrfsuite') since the exact parameters were not stated. The re-ranking of statements considering the identified labels was ambiguous, thus we keep appending sentences ordered by k-mixture-model as long as it does not exceed the desired length of summary. Additionally, the problems with NLTK are faced here too.

5 Results and Analysis

We applied all the summarization algorithms discussed in Sect. 4 to the 7,347 case judgments of the Indian Supreme Court (as stated in Sect. 3). All extractive text summarization approaches were executed on a LINUX 64 bit machine with 4 GB RAM and Core i5 processors. The abstractive neural network-based algorithm was executed on a Tesla K40c GPU with 12 GB RAM. The Online Resource[5] gives the summary generated by each method for a particular case judgment. We then performed two types of evaluation on the performance of the summarization algorithms - quantitative and qualitative.

5.1 Quantitative Evaluation

Following the traditional way of evaluating summaries, we compute ROUGE scores by comparing the algorithmically generated summaries with the gold standard summaries obtained from WestLaw. ROUGE scores measure the fraction of n-gram overlap with a reference summary. The ROUGE scores achieved by each method over the Indian Supreme Court case judgments are shown in Table 4. This table also reports the summary generation times of the different algorithms, for a particular case (one of the landmark cases used for the qualitative evaluation discussed below). LSA achieves the highest score in the family of classical unsupervised summarization techniques, followed by the method based on data reconstruction (DSDR). LexRank performed moderately, producing shorter sentences in the summary but its execution time was quite high. Frequency Summarizer (FreqSum) being a very naive method (only relies on frequency of words) performs poorly, though it takes the least time.

The neural network based unsupervised extractive method does not perform well from the perspective of ROUGE scores. But its execution time is lower than most of the others. The pointer-generator method for abstractive summarization perform moderately well in terms of ROUGE scores. As described in Sect. 4.1, we trained the abstractive model for 18,730 epochs (over five days). We observed that the performance gradually improves with more training; we plan to repeat the experiments with more training in future.

From the family of legal-specific summarization techniques, GraphicalModel and LetSum have comparable performance, while CaseSummarizer performs relatively poorly. This poor performance of CaseSummarizer is probably because

[5] Supplementary material, also available at https://drive.google.com/open?id=1KbcjdnvO1kHn3HNr1Jo-SI2XLbN72vD8.

Table 4. Performance of the different Text Summarization Approaches applied to Indian Supreme Court Judgments (US: unsupervised, S: supervised)

Broad class of Approaches	Methods from each Class	Type	ROUGE-1		ROUGE-2		ROUGE-L		Execution Time (sec)
			Recall	F-score	Recall	F-score	Recall	F-score	
Classical Extractive Summarization Approaches	LexRank	US	0.486	0.238	0.242	0.10	0.443	0.167	8.56
	LSA	US	**0.55**	**0.269**	**0.275**	**0.114**	**0.505**	**0.189**	2.43
	FreqSum	US	0.226	0.143	0.109	0.064	0.183	0.097	0.75
	DSDR	US	0.545	0.255	0.249	0.104	0.49	0.173	1.65
Legal Document specific Extractive Summarization	CaseSummarizer	US	0.198	0.139	0.094	0.063	0.154	0.094	7.95
	GraphicalModel	S	0.386	**0.351**	**0.171**	**0.159**	0.343	**0.297**	2.4
	LetSum	S	**0.408**	0.298	0.112	0.073	**0.371**	0.235	10.16
Neural Network based Suumarization	NeuralEx	US	0.138	0.198	0.055	0.076	0.125	0.132	1.09
	NeuralAbs	S	0.239	0.29	0.11	0.14	0.214	0.215	3.75 (GPU)

the approach is heavily dependent on the correct identification of named entities, which is difficult in case of legal documents using standard NLP tools (as discussed earlier).

5.2 Qualitative Evaluation

We gather opinion from legal experts on the quality of summaries generated by different algorithms. To this end, we chose three landmark cases well-known in Indian Law (see the Supplementary Information for details of the cases). For each case, we showed to the legal experts the gold standard summary (by Westlaw) and the summaries generated by some of the best performing algorithms according to the quantitative analysis described above. The summaries were annonymized, i.e., the experts were not told which summary was generated by which method. The experts decided to evaluate the summaries based on how well the summaries capture four important aspects of a case judgment – (1) the holding/ruling of the Court combined with the reasoning behind it, (2) the legal facts, (3) the statutes involved, and (4) precedents on which the judgments were based. They rated each summary on each aspect, on a Likert scale of 0–5 where 0 means the summary was poor and 5 means it was very good (in capturing the said aspect). The average ratings of the methods over the four aspects are shown in Fig. 1. Both our experts had high agreement on the scores given. Next, we give both the qualitative and the quantitative evaluation for understanding the trade-offs of using different algorithms.

WestLaw Gold Standard Summaries: The legal experts opined that the WestLaw summaries, though well-written, focus only on two aspects – facts and statute – while they do not cover the other two aspects well. The holding appeared at the end of the summary but the reasoning was not present. Two out of the three summaries evaluated did not contain precedents. Due to these limitations of the WestLaw summaries, some of the automatic methods we studied were given higher scores by the experts.

Latent Semantic Analysis (LSA): This algorithm achieves the maximum ROUGE recall and F-score among the classical unsupervised techniques.

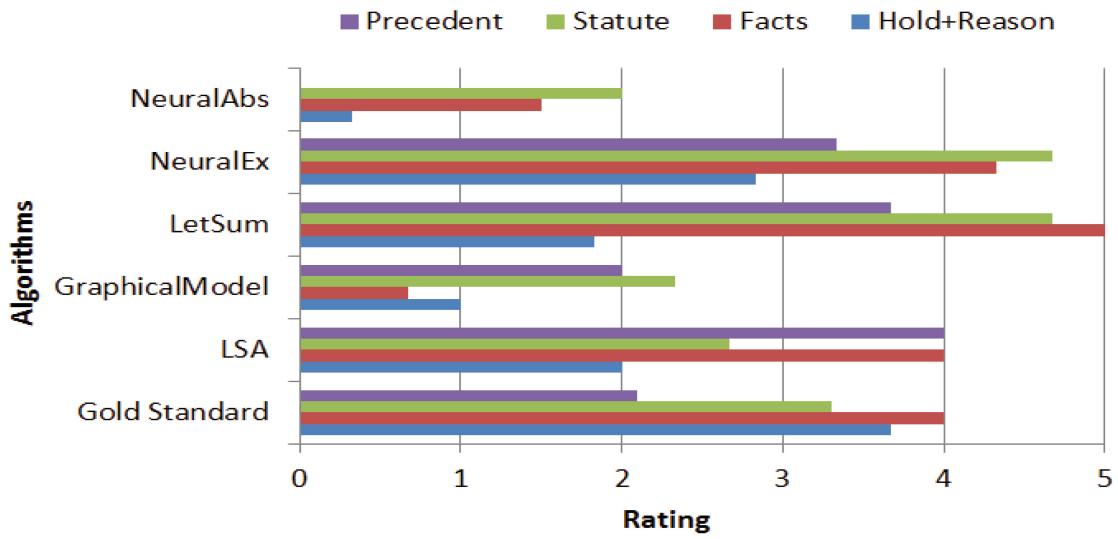

Fig. 1. Average ratings on a scale of 0–5 by legal experts of the summaries of three different cases generated by the algorithms

LSA achieves higher recall (but lesser F-score) when compared with the legal-specific summarization algorithms GraphicalModel and LetSum. The experts judged that initial parts of the summaries were nicely written and very relevant sentences were extracted. The facts of the case were presented well. Although LSA is an unsupervised approach, its inherent topic modeling features enables it to extract important sentences from each of the rhetorical categories. However, there are two limitations – (i) LSA has a tendency to pick lengthy sentences, and (ii) although the initial parts of the summary are good, the quality degrades drastically after covering around half of the document.

GraphicalModel: Graphical model achieves a comparable performance with LetSum w.r.t. ROUGE scores, while taking much lesser time. In fact, this method achieves the best F-score among all the extractive methods. However, according to the legal experts, the overall quality of the summary was not as promising as it seems from the ROUGE scores. GraphicalModel could extract well parts of the case where the statutes were quoted, and the arguments of the case. But the other two aspects were not reflected well in the summary.

LetSum: The performance of LetSum is comparable to GraphicalModel in terms of ROUGE scores. It has by far the highest execution time which is a drawback for online applications. According to the experts, the facts of the case and the parts of the case that described the statutes and precedents are covered well in the summaries. Like LSA, the initial part of the summary was good but the quality degrades gradually. The major drawback of this approach is its readability. As mentioned in their paper, LetSum extracts *textual units* and not complete sentences, which hampers the readability.

Neural Extractive (NeuralEx): This method does not perform well in terms of ROUGE score, though it performs slightly better than CaseSummarizer which

uses legal knowledge. Interestingly, the legal experts felt that the quality of the summaries was much better than that of all the other techniques. The summary has a high coverage, that is, it could extract sentences from all the rhetorical categories. Another important factor is that the execution time of this method is lesser than that of most other algorithms.

Neural Abstractive (NeuralAbs): The algorithm performs moderately in terms of ROUGE scores. But its disadvantage is in the running time – the training is resource intensive and the summary generation procedure is expensive. The summaries could partially represent the facts and statutes of the case. The other aspects did not occur much in the summary, simply because the reference WestLaw summaries on which the model was trained, did not have the other two aspects. It is possible that this method will perform better if trained over better quality summaries, but it is difficult to get good quality summaries in such high numbers as is necessary for training this model.

6 Concluding Discussion and Future Directions

In this paper, we have compared several text summarization approaches on legal case judgments from the Supreme Court of India. To our knowledge, this is the first systematic comparison of summarization algorithms for legal text summarization. We make the implementations of the algorithms explored in this paper at https://github.com/Law-AI/summarization. Our analysis leads to following insights.

(1) We understand that no one method can be considered as the best. While one method can best represent the facts of the case (LetSum), another might represent the statutes and precedents cited better (GraphicalModel). Simultaneously, in an online setting, the execution time is also a very important factor.
(2) None of the methods implemented could give the holding/ruling of the case combined with reasoning. This aspect is a very important part of the summary, because based on this a lawyer will decide whether or not to include the case as an argument in his favour.
(3) ROUGE scores might not always be the best evaluation metric to measure the quality of domain-specific summaries. ROUGE measures only n-gram overlaps and does not take into account whether the sentences represent all the facets of the document (e.g., rhetorical categories for legal documents).
(4) General summarization methods that require no knowledge of the domain may perform well quantitatively (LSA) and qualitatively (NeuralEx) but not both. Legal-specific summarization methods try to achieve the best of both the worlds. But their performances are highly dependent on manual annotations/gold-standard summaries for training and correct identification of domain-specific information in the text. An important future challenge is to develop a good and sufficiently large set of gold standard summaries for training supervised methods (especially neural models).

Acknowledgment. We sincerely acknowledge Prof. Uday Shankar and Uma Jandhyala from Rajiv Gandhi School of Intellectual Property Law, Indian Institute of Technology Kharagpur, India for their valuable feedback.

References

1. Australian Case Document. http://www.judgments.fedcourt.gov.au/judgments/Judgments/fca/single/2018/2018fca1517
2. Canadian Case Document. https://www.canlii.org/en/ca/fct/doc/2018/2018fc980/2018fc980.html
3. DSDR. https://gist.github.com/satomacoto/4248449
4. LSA and LexRank. https://pypi.python.org/pypi/sumy
5. UK Case Document. https://www.supremecourt.uk/cases/docs/uksc-2016-0209-judgment.pdf
6. Westlaw. https://legal.thomsonreuters.com/en/products/westlaw
7. Allahyari, M., et al.: Text summarization techniques: a brief survey. arXiv preprint arXiv:1707.02268 (2017)
8. Cao, Z., Wei, F., Li, S., Li, W., Zhou, M., Houfeng, W.: Learning summary prior representation for extractive summarization. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), vol. 2, pp. 829–833 (2015)
9. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 484–494 (2016)
10. Chopra, S., Auli, M., Rush, A.M.: Abstractive sentence summarization with attentive recurrent neural networks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93–98 (2016)
11. Das, D., Martins, A.F.: A survey on automatic text summarization. Lit. Surv. Lang. Stat. II Course CMU **4**, 192–195 (2007)
12. Dong, Y.: A survey on neural network-based summarization methods. CoRR abs/1804.04589 (2018). http://arxiv.org/abs/1804.04589
13. Erkan, G., Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Int. Res. **22**(1), 457–479 (2004)
14. Farzindar, A., Lapalme, G.: Legal text summarization by exploration of the thematic structure and argumentative roles. Text Summarization Branches Out (2004)
15. Farzindar, A., Lapalme, G.: Letsum, an automatic legal text summarizing system. Legal knowledge and information systems, JURIX, pp. 11–18 (2004)
16. Text summarization with NLTK (2014). https://tinyurl.com/frequency-summarizer
17. Galgani, F., Compton, P., Hoffmann, A.: Citation based summarisation of legal texts. In: Anthony, P., Ishizuka, M., Lukose, D. (eds.) PRICAI 2012. LNCS (LNAI), vol. 7458, pp. 40–52. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32695-0_6
18. Galgani, F., Compton, P., Hoffmann, A.: Combining different summarization techniques for legal text. In: Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data, pp. 115–123. Association for Computational Linguistics (2012)

19. Gelbart, D., Smith, J.: Beyond boolean search: flexicon, a legal tex-based intelligent system. In: Proceedings of the 3rd International Conference on Artificial Intelligence and Law, pp. 225–234. ACM (1991)
20. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: SIGIR, pp. 19–25 (2001)
21. Grover, C., Hachey, B., Hughson, I., Korycinski, C.: Automatic summarisation of legal documents. In: Proceedings of the 9th International Conference on Artificial Intelligence and Law, pp. 243–251. ACM (2003)
22. Grover, C., Hachey, B., Korycinski, C.: Summarising legal texts: sentential tense and argumentative roles. In: Proceedings of the HLT-NAACL 03 on Text Summarization Workshop, vol. 5, pp. 33–40. Association for Computational Linguistics (2003)
23. Gupta, V., Lehal, G.S.: A survey of text summarization extractive techniques. J. Emerg. Technol. Web Intell. **2**(3), 258–268 (2010)
24. Hachey, B., Grover, C.: Sentence classification experiments for legal text summarisation (2004)
25. Hachey, B., Grover, C.: Extractive summarisation of legal texts. Artif. Intell. Law **14**(4), 305–345 (2006)
26. He, Z., et al.: Document summarization based on data reconstruction. In: Proceedings of AAAI Conference on Artificial Intelligence, pp. 620–626 (2012)
27. Kågebäck, M., Mogren, O., Tahmasebi, N., Dubhashi, D.: Extractive summarization using continuous vector space models. In: Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC), pp. 31–39 (2014)
28. Kanapala, A., Pal, S., Pamula, R.: Text summarization from legal documents: a survey. Artif. Intell. Rev., 1–32 (2017). https://doi.org/10.1007/s10462-017-9566-2
29. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 159–165 (1958)
30. Agrawal, M., Mehta, P., Ghosh, K.: Overview of information access in legal domain fire 2013 (2013). https://www.isical.ac.in/~fire/wn/LEAGAL/overview.pdf/
31. Mihalcea, R., Tarau, P.: TextRank: bringing order into texts. In: EMNLP (2004)
32. Moens, M.F., Uyttendaele, C., Dumortier, J.: Abstracting of legal cases: the salomon experience. In: Proceedings of the 6th International Conference on Artificial Intelligence and Law, pp. 114–122. ACM (1997)
33. Nallapati, R., Zhai, F., Zhou, B.: Summarunner: a recurrent neural network based sequence model for extractive summarization of documents. In: AAAI, pp. 3075–3081 (2017)
34. Nallapati, R., Zhou, B., dos Santos, C., Gulcehre, C., Xiang, B.: Abstractive text summarization using sequence-to-sequence RNNS and beyond. In: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pp. 280–290 (2016)
35. Narayan, S., Cohen, S.B., Lapata, M.: Ranking sentences for extractive summarization with reinforcement learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), vol. 1, pp. 1747–1759 (2018)
36. Nenkova, A., McKeown, K.: A Survey of Text Summarization Techniques. In: Aggarwal, C., Zhai, C. (eds) Mining Text Data, pp. 43–76. Springer, Boston (2012). https://doi.org/10.1007/978-1-4614-3223-4_3
37. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304 (2017)

38. Polsley, S., Jhunjhunwala, P., Huang, R.: Casesummarizer: a system for automated summarization of legal texts. In: Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: System Demonstrations, pp. 258–262 (2016)
39. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 379–389 (2015)
40. Saravanan, M., Ravindran, B., Raman, S.: Improving legal document summarization using graphical models. In: Proceedings of the 2006 Conference on Legal Knowledge and Information Systems: JURIX 2006: The Nineteenth Annual Conference, pp. 51–60. IOS Press (2006)
41. See, A., Liu, P.J., Manning, C.D.: Get to the point: summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368 (2017)
42. Verma, S., Nidhi, V.: Extractive summarization using deep learning. arXiv preprint arXiv:1708.04439 (2017)
43. Yin, W., Pei, Y.: Optimizing sentence modeling and selection for document summarization. In: IJCAI, pp. 1383–1389 (2015)