ITC 3/53

Information Technology
and Control
Vol. 53 / No. 3/ 2024
pp. 772-784
DOI 10.5755/j01.itc.53.3.36602

A Hybrid Summarization Model for Legal Judgment Document Based on Domain Knowledge

Received 2024/03/09 | Accepted after revision 2024/05/29

HOW TO CITE: Song, Y., Huang, R., Chen, Y., Lin, C., Yu, S., Tang, R., Qin, Y. (2024). A Hybrid Summarization Model for Legal Judgment Document Based on Domain Knowledge. *Information Technology and Control*, 53(3), 772-784. https://doi.org/10.5755/j01.itc.53.3.36602

A Hybrid Summarization Model for Legal Judgment Document Based on Domain Knowledge

Yumei Song

Engineering Research Center of Text Computing & Cognitive Intelligence, Ministry of Education, Guizhou University, Guiyang, 550025, China;
College of Computer Science and Technology, Guizhou University, Guiyang, 550025, China;
College of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, 550025, China;
e-mail: songyumei123@163.com

Ruizhang Huang, Yanping Chen, Chuan Lin, Shuai Yu

Engineering Research Center of Text Computing & Cognitive Intelligence, Ministry of Education, Guizhou University, Guiyang, 550025, China;
College of Computer Science and Technology, Guizhou University, Guiyang, 550025, China

Ruixue Tang

School of Information, Guizhou University of Finance and Econnomics, Guiyang, 550025, China

Yongbin Qin

Engineering Research Center of Text Computing & Cognitive Intelligence, Ministry of Education, Guizhou University, Guiyang, 550025, China;
College of Computer Science and Technology, Guizhou University, Guiyang, 550025, China;
e-mail: ybqin@gzu.edu.cn

Corresponding authors: Yongbin Qin ybqin@gzu.edu.cn, Yumei Song songyumei123@163.com

Legal judgment document summarization, as a task specific to the legal domain, involves automatically generating a fluent, informative, and well-organized summary from the original legal judgment document. Unlike traditional text summarization tasks, this domain-specific task places higher demands on content accuracy and completeness in the summary, while also requiring the preservation of the professional expression found in the original text. Consequently, conventional summarization methods often struggle to perform effectively in the legal domain. In response to this challenge, this paper introduces a hybrid summarization model tailored for legal judgment documents. Our model harnesses the strengths of both extractive and abstractive summarization methods, incorporating domain knowledge to enhance the summary generation process. We conduct extensive experiments to verify the effectiveness of our proposed method and compare the results with a baseline using ROUGE evaluation metrics. The experimental findings highlight that our model excels in providing more accurate and readable summarizations compared to traditional methods.

KEYWORDS: Legal summarization, domain knowledge, pointer-generator network, hybrid model, text summarization.

1. Introduction

In recent years, with the development of smart justice and the increasing number of digitized legal judgment documents, automatic legal judgment document summarization has gained increasing attention. Legal judgment document summarization [17] serves as a domain-specific automatic text summarization [1, 21] task to automatically identify the important information of a legal judgment document and express it in a human-readable summary. It enables legal practitioners to quickly understand an expatiatory legal judgment document. However, legal judgment documents tend to be long and full of professional expressions, making this task even more challenging.

There are two types of traditional text summarization methods: extractive and abstractive. Extractive methods retrieve significant sentences or keywords from the original document in order to construct a summary. These methods tend to copy sentences directly, do not accurately summarize the original meaning, and have low readability. Conversely, abstractive methods have the capability to generate novel terms in order to succinctly encapsulate the content of the original document. These methods are better at expressing the original meaning, but tend to generate hallucinations [3] and are not suitable for long texts.

In fact, a direct application of traditional summarization methods to the task of legal judgment summarization is not realistic. The main reasons are as follows: 1) Traditional summarization methods do not have enough capacity to capture the important domain-related information of the original document, and therefore cannot guarantee the accuracy and completeness of the summary. Traditional methods typically generate summaries from textual features such as word frequency and word position without any external domain knowledge constraints, so that important domain-related information is easily lost. For example, as shown in Figure 1, the sentence underlined is easy to ignore in traditional methods if without any domain constraints. But in fact, this sentence is clearly crucial to the document as it describes important case facts. Thus, the crux of the matter we need to confront is how to leverage domain knowledge to enhance the model's ability to capture important domain-related information. 2) Legal judgment documents contain many professional and lengthy expression. We expect to generalize it in a more concise way, in particular with some specialized terms. Traditional summarization methods, either extractive or abstractive methods, generate summaries from input documents or general vocabulary, and that are easy to cause out-of-vocabulary (OOV) problems. Furthermore, without domain knowledge guidance, it is hard to generate novel and accuracy words to generalize the documents. To conclude, it is imperative to incorporate domain knowledge in order to provide guidance for the process of generating summaries due to the aforementioned reasons.

In this work, we propose a hybrid legal judgment summarization model (HLSum) based on domain knowledge. Firstly, a knowledge-aware extractor is devised to identify the significant sentences from legal judgment documents based on the domain knowledge. The knowledge-aware identify extractor (K-Extractor) can roughly extract important information that is not only semantically but also domain-specific. Moreover, in this part, it can greatly reduce the length of the document. Secondly, we propose a knowledge-oriented pointer-generator network (K-PGN) to generate the final summary based on the selected significant sentences by the K-Extractor. By incorporating domain knowledge into the pointer-generator network, K-PGN enhances content accuracy and mitigates out-of-vocabulary (OOV) word issues. The unique feature of the K-PGN is that it helps to generate more concise and accurate expressions, to obtain a more abstractive summary. Overall, our contributions in this paper are as follows: 1) We propose a hybrid legal judgement summarization model that incorporates domain knowledge. The consideration of domain knowledge for the summary makes it difficult to lose or miss important domain-related information and enhances the consistency between the summary and the original document to a certain extent. To our best understanding, our work represents an initial endeavor towards this task. 2) We propose a novel knowledge-oriented pointer-generator network (K-PGN) based on domain knowledge. The model can generate a more readable and concise summary, and mitigates out-of-vocabulary (OOV) word issues very well. 3) The effectiveness of the model is verified by a large number of experiments. Experimental results show significant improvements in summarization performance on several evaluation metrics.

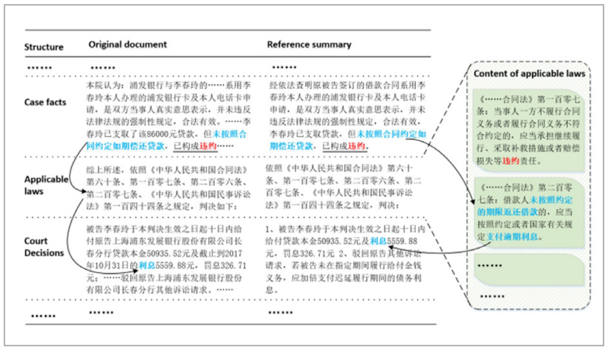
2. Related Work

The ultimate goal of text summarization is to produce a concise and coherent overview that encapsulates the essential points of the original document, minimizing redundancy and maximizing the utilization of limited space. Two overarching approaches to text summarization are extractive and abstractive.

The extractive approach involves three key steps: 1) Creating a suitable representation of the original text. 2) Scoring each sentence based on this text representation. 3) Extracting sentences with high scores and concatenating them to form the summary. Several works have been conducted in recent years on extractive summarization, employing various methodologies, including: statistical and semantic features approaches [16, 23, 39], probabilistic approaches [4, 9], graph-based approach [24, 30], traditional machine learning based approach [35], neural network-based approach [22, 28, 31].

Figure 1

The domain knowledge is important to the legal judgment summarization. The contents marked in blue in the original document is related with the content of second applicable law, and the summary also need to maintain those contents. The terminology in red appears only once in original document, and it also appears in the first applicable law, make it appear in the summary as important information



Abstractive summarization has long been regarded as a challenging task. However, recent years have witnessed significant advancements in this area, particularly due to the impact of the rapid development of neural networks. Rush et al. [32] pioneered the use of a neural attention seq2seq model for abstractive summarization. Nallapati et al. [27] introduced an RNN encoder-decoder architecture tailored for summarization tasks. Drawing inspiration from the pointer mechanism proposed by [38], See et al. [33] presented a pointer-generator network to address challenges associated with rare words and out-of-vocabulary (OOV) terms. Sun et al. [36] put forth a novel multisource pointer network for product title summarization, incorporating a new knowledge encoder to enhance pointer network performance. Wang et al. [40] introduced a concept pointer network for abstractive summarization. Presently, the pointer-generator network has become the mainstream method for abstractive summarization due to its commendable performance.

Recently, there has been a marked focus on legal document summarization, resulting in noteworthy achievements. One seminal work, Grover et al. [14]

described a legal corpus comprising 188 judgments from the House of Lords Judgment (HOLJ) website from 2001–2003, specifically for extractive summarization of British judgments. Classical algorithms like LexRank [7], Latent Semantic Analysis (LSA) [18, 25] and TextRank [24] have been widely applied in the legal domain. However, due to the unique nature of legal documents, the performance of these methods is not satisfactory. Galgani et al. [10, 12] proposed a citation-based summarization method to generate catchphrases from citation text or use citations to select sentences from original document. This method is limited to the Anglo-American law system and may not be applicable to civil law systems. Hachey et al. [15], Ghosh et al. [13] align different sentences associated with rhetorical roles in final summary generation. Rhetorical roles act as valuable information, enhancing the readability and coherence of the final summary. Galgani et al. [11] applied a knowledge base (KB), created based on the ripple-down rules of Compton and Jansen (1990), to generate summaries by combining different summarization techniques. While these legal summarization methods have achieved some effectiveness, most do not deeply consider the domain knowledge of legal documents, primarily focusing on the Anglo-American law system. Therefore, there is a need to study a more efficient legal summary method based on domain knowledge, especially for civil law systems.

In legal domain, automatic summarization is different from it in general because legal judgment documents often have a special internal structure and contain a lot of domain knowledge. The internal structure of judgment document depends upon the country of the case, for example, the Chinese judgment document often consists of header, main body, court decisions and tail. The main body is the core of the judgment documents, requires a clear description of the facts and evidence involved in the case, especially the facts ascertained by the court, it usually includes plaintiff and defendant information, plaintiff's appeals, case facts, judicial evidences and court opinions. The court decisions describe on what statute does the court decide whether or not the plaintiff's claim should be upheld. The legal summarization is mainly generated by the main body and court decisions. More importantly, the legal documents contain a lot of domain knowledge, such as applicable law, judicial interpretation, trial guidance and so on. In this article, we choose applicable laws as domain knowledge. Because the applicable law plays a very important role in legal judgment documents. For example, the applicable laws of the case rely on the case facts, and the applicable laws of the case affect the court decisions, as shown in Figure 1. In addition, the content related to the applicable law in the original text is generally important that needs to be retained in the abstract. Thus, we can make full use of the internal structure and applicable laws of judgment document to promote the performance of legal summarization.

3. Our Model

We propose a novel hybrid legal summarization model composed of a knowledge-aware extractor and a knowledge-oriented pointer-generator network. Our model leverages domain knowledge, specifically the applicable laws from the original legal judgment documents, to enhance the performance of the summarization task. Prior to delving into the specifics of our model, we establish the roles and tasks of both the knowledge-aware extractor(K-Extractor) and the knowledge-oriented pointer-generator network (K-PGN).

Problem Definition. Let $\boldsymbol{d}$ denote a judgment document containing k sentences $\boldsymbol{d}=\{sent_1, sent_2, \ldots, sent_k\}$, where the $sent_i$ is the $i\text{-}th$ sentence in the document $\boldsymbol{d}$. We filter out irrelevant sentences based on the internal structure of legal judgment document, and get sentence set $\{sent_1, sent_2, \ldots, sent_m\}$ as the input of knowledge-aware extractor.

The knowledge-aware extractor can be defined as a task of assigning a label l_i to each $sent_i$ where $i \in [1,m]$, indicating whether the sentence is so important that it is suitable as the input of K-PGN.

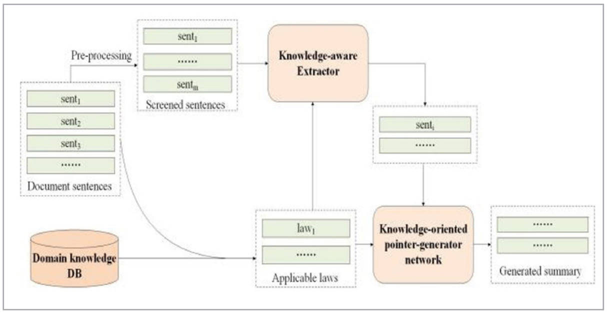
The input of K-PGN is a sequence of words $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$,where n is the word index. The output of the K-PGN is the finial summary sequence $\mathbf{y} = \{y_1, y_2, \ldots, y_t\}$, where the y_i is the $i\text{-}th$ word of the final summary.

The structure of our model is shown in Figure 2.

Pre-processing. Considering the special structure of the judgment document, we first filter the sentences of the input legal judgment document, and remove the sentences that are obviously irrelevant to the gen-

Figure 2

The structure of our model



eration of the summary. By comparing a large number of legal summaries, we found that a qualified summary should include: case type, plaintiff's appeals, facts confirm by the court, court opinions, applicable laws and court decisions, show as in Figure 1. So, we use simple method such as regular expression to pick these parts that are useful for summary generation. We remove sentences that are obviously irrelevant to the summary, such as the head of the judgment document, the background information of the identity of the original defendant, the information of the hearing process of the case, the tail, and the explanation of the case. Through pre-processing, the input is greatly shortened as $\{sent_1, sent_2, \ldots, sent_m\}$, $m<k$, which not only reduces the computational complexity, but also avoids the noise caused by irrelevant information.

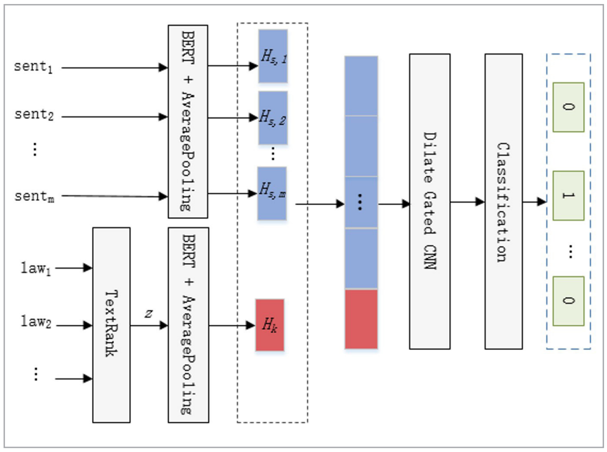
3.1. Knowledge-aware Extractor

The first part of our model is a knowledge-aware extractor (K-Extractor), as shown in Figure 3. Differently, our knowledge-aware extractor does not need to obtain the final summary, but just needs to obtain a short list of sentences with high information to further facilitate the K-PGN. The principle of extractor is to seek completeness, that is, try to cover the information required by the final summary as much as possible. Thus, we treat this task as sequence labeling problem with the unit of sentence. In addition, by using an extractor, the text length can be greatly shortened without losing important information, thus solving the problem that legal texts are usually very long, which makes summary generation difficult.

Unlike general domains, legal domains calculate the importance of each sentence by considering not only its general semantic features, but also its domain-related information. We use the applicable laws' content as domain knowledge to assist in picking the important sentences. Each judgment document has one or more applicable laws, which are strongly related to the case facts and the court decisions. Hence, we build a domain knowledge base of laws which include all laws that may appear in legal judgment documents. This domain knowledge base lists the specific content of all the laws, as shown in Figure 1. In addition, regular expression and other simple extraction methods are used to obtain the applicable laws $\{law_1, law_2, \ldots\}$ in legal judgment document $\boldsymbol{d}$. These applicable laws all apply to the same case, so there is a certain correlation between them. With these reasons in mind, we can extract the key information of these laws, so that it can better help to obtain the domain-related sentences in legal judgment documents. We use a TextRank [26] model to obtain the key words of all applicable laws of the original legal judgment document, and then connect all the key words as one sentence z.

Figure 3

The architecture of knowledge-aware extractor



Next, we use BERT [6] and average pooling to get the representation of the sentence z, as shown in Equation (1). Similarly, we use BERT and average pooling to get the representation of each individual sentence $sent_i$ where $i \in [1,m]$ in the legal judgment document, as shown in Equation (2). In order to select more domain-related sentences, we spliced the applicable laws and sentences of legal judgment documents to increase the domain-related knowledge in the original text and greatly increase the probability of domain-related sentences being selected. We concatenate the two above representations to add domain-related information and increase the probability that a domain-related sentence will be selected, as shown in Equation (3).

$$H_k = avgpool(Bert(z)) \tag{1}$$

$$H_{s,j} = avgpool(Bert(sent_i)), \forall i \in [1, m] \tag{2}$$

$$X = concat(H_{s,1}, \cdots, H_{s,m}, H_k) \tag{3}$$

Then, a Dilate Gated Convolutional Neural Network (Dilate Gated CNN) [34], which integrates expansion convolution and gate convolution, be used to learn the semantic representation of sentences.

$$Y = X + D_1(X) \otimes \sigma\big(D_2(X)\big), \tag{4}$$

where σ is a sigmod function, and D_1 and D_2 are dilate convolutional neural network with different parameters. We use a classification to get the label of each sentence.

3.2. Knowledge-oriented Pointer-generator Network

The second component of our model is a knowledge-oriented pointer-generator network that generates the summary word-by-word. In this section, we have enhanced the pointer generation network proposed by See et al. [33], incorporating domain knowledge.

Encoder-decoder model. The encoder-decoder model consists of a two-layer bidirectional LSTM-RNN encoder and a one-layer unidirectional LSTM-RNN decoder, introduced with an attention mechanism. The input word sequence, denoted as $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, is processed by the encoder, mapping the text into a sequence of encoder hidden states $\{h_1, h_2, ..., h_n\}$. During each decoding time step t, the decoder takes the previous word embedding and the preceding context vector as input to compute the decoder hidden state s_t. The generation of the target summary from a vocabulary probability distribution $P_{vocab}(w)$ follows this process:

$$P_{vocab}(w) = P(y_t|y_{<t}, x; \theta) = softmax(W_2(W_1[s_t, c_t] + b_1) + b_2). \tag{5}$$

Here, s_t represents the context vector at time step t, and W_2, W_1, b_1, b_2 are trainable parameters. The context vector c_t is a weighted sum of h_i of the input text, with weights determined by the attention mechanism $a_{t,i}$.

The attention weights $a_{t,i}$ are computed using the softmax function with learnable parameters v, W_h, W_s, b:

$$c_t = \sum_{i=1}^{n} a_{t,i} h_i \tag{6}$$

$$a_{t,i} = softmax\big(v \tanh(W_h h_i + W_s s_t + b)\big). \tag{7}$$

Pointer generator network. The Pointer Network, introduced by Vinyals et al. [38], utilizes the attention mechanism [2] as a pointer to choose words from the input instead of choosing from a fixed vocabulary, making it particularly suitable for extractive summarization. The Pointer Generator Network consists of two parts: one utilizes pointer to choose words from the input, and the other picks new words from one fixed vocabulary. These two parts work together to jointly figure out the probabilities of the words in final summary. The generation probability $p_{gen} \in [0,1]$ of the pointer generation network [33] can be obtained by the following equation:

$$P_{gen} = \sigma\big(W_c c_t + W_s s_t + W_y y_{t-1} + b_{ptr}\big). \tag{8}$$

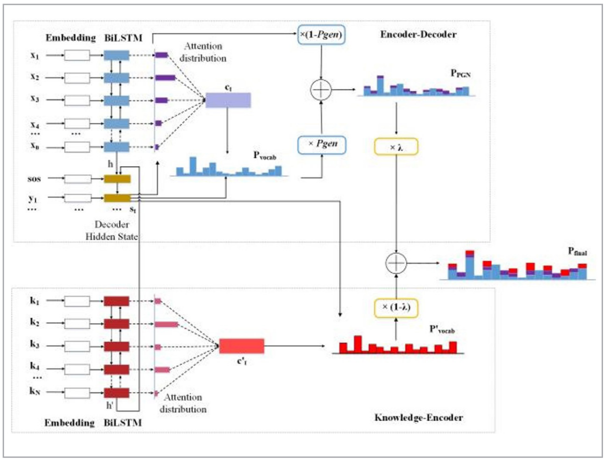
Here, σ is a sigmoid function, and the vectors W_c, W_s, W_y along with the scalar b_{ptr} are learnable parameters. The p_{gen} serves as a switch pointer to pick a word from a fixed vocabulary or the input sequence. Consequently, the probability distribution $P_{PGN}(w)$ is given by:

$$P_{PGN}(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i = w} a_{t,j}. \tag{9}$$

It is important to note that P_{vocab} is zero for a word w is an out-of-vocabulary word.

Knowledge Pointer generator. As the legal document has strong domain characteristics, the direct utilization of pointer-generator network may ignore the important implicit information in source text, leading to degrade the performance of our model. Therefore, we use a knowledge encoder to encodes the applicable laws' content which involved in each legal judgment document. The knowledge encoder uses a bidirectional LSTM to encode the contents of laws, which is similar to the Encoder in the pointer-generating network. The content of applicable laws is taken as input, and an intermediate vector is calculated, through which a global dictionary probability distribution $p'_{vocab}(w)$ can be obtained. Finally, the probability distribution $p_{PGN}(w)$ calculated by the pointer generation network is fused to $p'_{vocab}(w)$ to obtain the final probability distribution $p_{final}(w)$. Through this knowledge encoding method, the information related to applicable laws can be strengthened in the source document, so as to improve the probability of accurate word generation, and further improve the performance of the whole legal summary generation. The architecture of the knowledge-oriented pointer-generator network is shown in Figure 4.

Figure 4
The knowledge-oriented pointer-generator network



The model combines the source document and the applicable laws of source document to produce summary. The input source $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ and the applicable laws $\mathbf{K} = \{k_1, k_2, ..., k_m\}$ all encode by a bidirectional LSTM, and then get a series of hidden states $(h_1, h_2, ..., h_N)$ and $(h'_1, h'_2, ..., h'_M)$. After that, the initial state s_0 of decoder is obtained by connecting and converting the last hidden state h_N and h'_M of the two encoders.

$$s_0 = ReLU(W_f[h_N, h'_M]), \tag{10}$$

where the ReLU=max(0,x), W_f is a learnable parameter.

Be similar to the Equation 7, attention weight distribution of the applicable laws a'_t can be calculated.

$$a'_{t,i} = softmax(v'tanh(W'_h h'_i + W'_s s'_t + b')), \tag{11}$$

where the W'_h, W'_s, v', b' is a trainable parameter. The s_t represents the hidden state of decoder at time step t and is computed like:

$$s_t = f(s_{t-1}, y_{t-1}, c_{t-1}, c'_{t-1}), \tag{12}$$

where s_{t-1} is the hidden state of decoder at time step t-1, y_{t-1} is the input of decoder at time step t, f represents a nonlinear function, and this paper adopts LSTM as function f. The c'_{t-1} is the context vector of the applicable laws at time step t-1 and is computed like:

$$c'_t = \sum_{i=1}^{n} a'_{t,i} h'_i. \tag{13}$$

Finally, the probability distribution $p_{PGN}(w)$ calculated by the pointer generation network is fused to $p'_{vocab}(w)$ to obtain the final probability distribution $p_{final}(w)$.

$$P_{final}(w) = \lambda P_{PGN}(w) + (1 - \lambda) P'_{vocab}(w), \tag{14}$$

where the generation probability λ is learned by:

$$\lambda = \sigma(W_c c_t + W_s s_t + W_y y_{t-1} + W_{c'} c'_t), \tag{15}$$

where vector W_c, W_s, W_y, $W_{c'}$ are learnable parameters. The training loss function for our model is as follows:

$$loss = \frac{1}{T} \sum_{t=0}^{T} - log\, P_{final}(w). \tag{16}$$

4. Experiment

4.1. Dataset Construction

This paper uses the dataset provided by the challenge of AI (CAIL2020, https://github.com/china-ai-law-challenge/CAIL2020/tree/master/sfzy) legal summarization track, which is the first dataset of legal summarization in China. It contains 4,047 marked civil judgment documents, involving nine causes of action, including labor contract, tort liability, lease contract, loan contract, inheritance, right of recourse, loan, infringement and inheritance relation. According to statistics, there are 36 kinds of laws involved in the dataset. After data cleaning, the data sets are divided according to 6:2:2, and 2340 judgment documents in training datasets, 779 judgment documents in verification datasets and 785 judgment documents in test data sets. The maximum number of words in the judgment documents is 13,064 and the minimum number is found 866 words, with an average of 2568 words.

After obtaining the statistical data of applicable laws in the dataset, we crawl the applicable laws from the network and construct a law library as domain knowledge base. In the data pre-processing stage, the specific content of the applicable laws involved in each judgment document is taken as domain knowledge.

4.2. Experiment Settings

In this paper, PyTorch framework is used to build the model. For the Encoder end of judgment documents and laws, 512 dimension bi-directional LSTM is used, while for the Decoder end, 512 dimension unidirectional LSTM is used. In terms of word vector, this paper adopts the method of random initialization, and sets the dimension of word vector as 512 dimension, which will be adjusted continuously in the process of continuous training. In addition, in the construction of dictionary, through the joint statistics of judgment documents and laws. In the whole process of training and testing, the input text length is compressed after data pre-processing. Therefore, the maximum length of the input source text in this paper is set to 700, which can effectively meet the requirements of the model and data after statistics. The maximum length of the generated text summary is set to 300, and the maximum length of the external knowledge of the law is set to 100. In this paper, the learning rate is set to 0.001, the initial value of the accumulator is set to 0.1, and the batch size of the training is 32. The generic ROUGE evaluation index is used for performance evaluation.

4.3. Evaluation Metrics

We use ROUGE [20] as the evaluation metric to evaluate our model. It evaluates the quality of a generated summary by calculating the overlap of lexical elements between a candidate summary and a reference summary, such as n-grams. Following established conventions, we have opted for the metrics ROUGE-1, ROUGE-2, and ROUGE-L, which individually assess the word overlap, bigram overlap, and the longest common sequence between the reference summary and the generated summary. We focus on the F-1 scores of ROUGE-1, ROUGE-2, and ROUGE-L. The computation is expressed as follows:

$$\mathrm{ROUGE-N} = \frac{\sum_{S\in \mathrm{RefSummary}} \sum_{\mathrm{gram}_n \in S} \mathrm{Count}_{\mathrm{match}}(\mathrm{gram}_n)}{\sum_{S\in \mathrm{RefSummary}} \sum_{\mathrm{gram}_n \in S} \mathrm{Count}(\mathrm{gram}_n)}. \quad (17)$$

4.4. Baseline

To validate the effectiveness of our proposed model, we conducted a comparative analysis against baselines. **Lead-3**, which is a classical extractive model in journalism, picks the first three sentences of an article to form the summary. Leveraging the common observation that crucial news information often resides in the initial portion of an article, the Lead-3 algorithm tends to yield favorable results. **TextRank** [26] is a keyword extraction framework that calculates the scores of the keywords or sentences in the text according to a PageRank-like algorithm, and then selects the words or sentences with the highest scores to build the summary. **BertSum** [22] represents a simplified variant of BERT tailored for extractive summarization. Following the original paper, we employed classifiers, transformers, and RNNs as the classification layer. **Seq2seq+att** utilizes a two-layer BiLSTM encoder and a one-layer LSTM decoder with attention mechanisms. **BART** [19] is a cutting-edge natural language processing model that builds upon the transformer architecture. It is applicable to natural language generation, translation, and comprehension. **Pointer-Generator Network (PGN)** [33] is a hybrid model that combines Seq2Seq-Gen with a pointer network. PGN can not only generate words from a fixed vocabulary, but also copy words from the input.

5. Results and Analyses

This section compares the performance of our proposed model with various benchmark algorithms. Table 1 shows the comparative results for ROUGE-1, ROUGE-2, and ROUGE-L on the CAIL2020 dataset. Figure 5 provides a visual representation of the comparison results among different methods. Our hybrid model not only generates an abstractive summary but also extracts crucial sentences from legal documents. The model exhibits substantial improvements, establishing a new state-of-the-art in both extractive and abstractive methods.

The improvement rates of the proposed method were calculated using Equation (18) with ROUGE-1, ROUGE-2, and ROUGE-L metrics.

$$Improved\_rate = \frac{P_{method} - C_{method}}{C_{method}} \times 100. \tag{18}$$

Here, p_{method} represents the proposed method, c_{method} denotes the compared method, and the results are presented in Table 2.

In this section, we evaluate the results of our model in comparison with the extractive model discussed in Section 5.1 and the abstractive model discussed in Section 5.2. To verify the effectiveness of our work, we conduct an ablation study in Section 5.3. Additionally, we perform human evaluation in Section 5.4 to assess the relevance, readability, and consistency of the generated summaries. Furthermore, in Section 5.5, we present a case study that demonstrates our model's capability to provide superior abstractive summaries compared to other baselines.

Table 1

Comparison of Proposed Approaches

Method	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1
Lead-3	1.00	0.12	1.97
TextRank	37.10	18.20	31.03
BertSum+class	30.04	11.18	20.89
BertSum+trans	31.86	12.59	22.52
BertSum+rnn	29.58	10.95	20.78
Seq2seq+attn	41.16	19.82	36.73
BART	49.38	30.53	43.14
PGN	46.36	21.55	39.26
our model	**57.12**	**35.09**	**54.37**

5.1. Extractive Summarization

In this extractive paradigm, we compare our model with several extractive mode, such as Lead-3, TextRank, and BertSum. From Table 1 and Table 2, we can see that the performance of Lead-3 is very poor, because this model only picks the first 3 sentence as summary. It also shows that the methods used in the field of journalism are not applicable in the legal field. Other general-domain methods, TextRank and BertSum, also do not perform well. The data in Table 2 illustrates substantial performance enhancements achieved by the proposed method when compared to TextRank and BertSum+trans. Specifically, our method improves ROUGE-1 scores by 53.96% for TextRank and 79.28% for BertSum+trans. Furthermore, there are significant improvements in ROUGE-2 scores, with enhancements of 92.80% for TextRank and 178.71% for BertSum+trans. Additionally, the proposed method outperforms in ROUGE-L, showing improvements of 75.22% for TextRank and 141.43% for BertSum+trans. These findings underscore the effectiveness of our approach in elevating summarization performance across various evaluation metrics and models. The score of ROUGE-2 and ROUGE-L improve more than that of ROUGE-1. The addition of domain knowledge allowed the final summary to include more judicial terminology and judicial specific expressions, resulting in a significant increase in ROUGE-2 and ROUGE-L scores, which was also more realistic.

Table 2

Improvement obtained by proposed method (%)

Method	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1
Lead-3	5612.00	29141.67	2659.90
TextRank	53.96	92.80	75.22
BertSum+class	90.15	213.86	160.27
BertSum+trans	79.28	178.71	141.43
BertSum+rnn	93.10	220.46	161.65
Seq2seq+attn	38.78	77.04	48.03
BART	15.67	14.94	20.03
PGN	23.21	68.83	38.49

5.2. Abstractive Summarization

We compare our model with tree representative abstractive model, Seq2seq, BART and PGN, and our abstractive model is an improvement on PGN.

The results in Tables 1-2 highlight the significant performance improvements achieved by the proposed method across different summarization models. Specifically, our method enhances the performance in Seq2seq, BART, and PGN on ROUGE-1 by 38.78%, 15.67% and 23.21%, respectively. Similarly, for ROUGE-2, there are improvements of 77.74%, 14.94% and 68.83% in Seq2seq, BART, and PGN, and for ROUGE-3, improvements of 48.03%, 20.03% and 38.49%, respectively. These findings underscore the effectiveness of our proposed method in significantly enhancing summarization performance across various evaluation metrics and models.

5.3. Ablation Study

In this section, we investigate the impact of the strategies proposed in the paper on model performance by conducting ablation experiments. Specifically, we explore the influence of domain knowledge and the hybrid model separately. The results of these ablation experiments are presented in Table 3. Upon comparison of the results between TextRank and K-Extractor, as well as PGN and K-PGN, it is evident that the incorporation of domain knowledge significantly enhances the summarization performance. Furthermore, our hybrid model demonstrates at least a two-point improvement over K-Extractor and K-PGN. This observation underscores the effectiveness of combining extractive and abstractive methods in improving summarization performance.

Table 3
Ablation experiments

Method	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1
TextRank	37.10	18.20	31.03
K-Extractor	53.43	33.97	46.06
PGN	46.36	21.55	39.26
K-PGN	55.01	32.03	50.98
K-Extractor+PGN	51.32	31.92	49.75
Hybrid model	57.12	35.09	54.37

5.4. Human Evaluation

To evaluate the relevance, readability and consistency of the summaries, we also performed a human evaluation. Relevance evaluates whether the summary includes crucial information from the original document while avoiding irrelevant and redundant details. Readability is based on the fluency, grammaticality, and coherence of the summary. Consistency assesses whether the content described in the summary aligns with the original document, avoiding contradictory and inaccurate descriptions. We compared the results of our model and the pointer-generator network on those tree human evaluation metrics.

To do human evaluation, we chose 100 samples from the test set randomly and enlisted three human evaluators for each sample. The evaluators scored each summary on the three metrics using a scale of 1 to 3 (3 for good, 2 for moderate, and 1 for bad). The average scores from the three evaluators for each summary were calculated. The results, shown in Table 4, indicate that our model outperforms the pointer-generator network across all three metrics, with a notable improvement in the consistency metric.

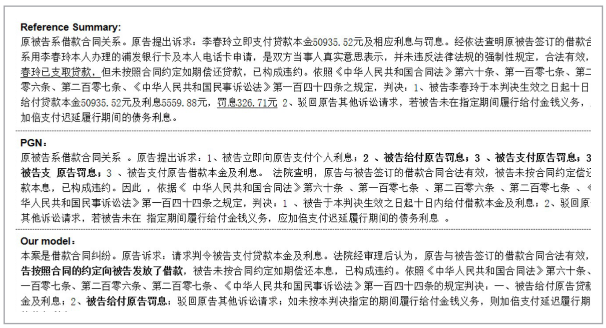
Table 4
Human Evaluation: comparison between our model and pointer-generator network

Method	Relevance	Readability	Consistency
PGN	2.07	2.29	1.40
our model	**2.14**	**2.76**	**2.08**

5.5. Case Study

To assess the performance of our model in real-case scenarios, we selected authentic samples from the CAIL2020 dataset. As depicted in Figure 5, the summaries produced by the Pointer-Generator Network (PGN) exhibit significant duplication (highlighted in bold), with crucial content, such as the underlined portion in the reference summary, being omitted. In contrast, our model addresses the deficiencies in the PGN-generated summary by introducing missing content (highlighted in bold) and preserving the essential elements of the reference summary. Moreover,

Figure 5
Human Evaluation: comparison between our model and pointer-generator network


Reference Summary:
原被告系借款合同关系。原告提出诉求：李春玲立即支付贷款本金50935.52元及相应利息与罚息。经依法查明原被告签订的借款合
系用李春玲本人办理的浦发银行卡及本人电话卡申请，是双方当事人真实意思表示，并未违反法律法规的强制性规定，合法有效，
春玲已支取贷款，但未按照合同约定如期偿还贷款，已构成违约。依照《中华人民共和国合同法》第六十条、第一百零七条、第二
零六条、第二百零七条、《中华人民共和国民事诉讼法》第一百四十四条之规定，判决：1、被告李春玲于本判决生效之日起十日
给付贷款本金50935.52元及利息5559.88元，罚息326.71元 2、驳回原告其他诉讼请求，若被告未在指定期间履行给付金钱义务，
加倍支付迟延履行期间的债务利息。

PGN：
原被告系借款合同关系 。原告提出诉求：1、被告立即向原告支付个人利息；**2 、被告给付原告罚息；3 、被告支付原告罚息；3**
被告支 原告罚息；3 、被告支付原告借款本金及利息。 法院查明，原告与被告签订的借款合同合法有效，被告未按合同约定偿
款本息，已构成违约。因此 ，依据《 中华人民共和国合同法》第六十条 、第一百零七条 、第二百零六条 、第二百零七条 、《
华人民共和国民事诉讼法》第一百四十四条之规定，判决：1 、被告于本判决生效之日起十日内给付借款本金及利息；2、驳回原
其他诉讼请求，若被告未在 指定期间履行给付金钱义务，应加倍支付迟延履行期间的债务利息 。

Our model：
本案是借款合同纠纷。原告诉求：请求判令被告支付贷款本金及利息。法院经审理后认为，原告与被告签订的借款合同合法有效，
告按照合同的约定向被告发放了借款，被告未按合同约定如期偿还本息，已构成违约。依照《中华人民共和国合同法》第六十条、
一百零七条、第二百零六条、第二百零七条、《中华人民共和国民事诉讼法》第一百四十四条的规定判决：一、被告给付原告贷
金及利息；2、**被告给付原告罚息**；驳回原告其他诉讼请求；如未按本判决指定的期间履行给付金钱义务，则加倍支付延迟履行期

the summaries generated by our model are well-organized and more readable. In summary, our model excels in retaining intricate judicial details, resulting in a more comprehensive and coherent summary context than the PGN.

6. Conclusion

In the presented article, we introduce a hybrid model that leverages the advantages of both extractive and abstractive summarization methods for the summarization of legal judgment documents. This model incorporates domain knowledge, utilizing it to enhance the generation of legal summaries. To assess the efficacy of our approach, we conducted numerous comparative experiments against baseline methods. The results of these experiments reveal that our proposed method demonstrates superiority over existing techniques. Furthermore, our model effectively addresses the challenge of summarizing lengthy legal documents, a problem that has been difficult to tackle with other approaches.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grant No.62066008 and NO.62066007. This work is also supported by Scientific Research Project of Higher Education Institutions of Guizhou Province (Youth Project) (Guizhou Education Technology (2022) No.149). This work is also supported by Guizhou Provincial Science and Technology Projects under Grant No. ZK[2022]027.

References

1. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., Kochut, K. Text Summarization Techniques: A Brief Survey. arXiv Preprint arXiv:1707.02268, 2017. https://doi.org/10.14569/IJACSA.2017.081052
2. Bahdanau, D., Cho, K., Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv Preprint arXiv:1409.0473, 2014.
3. Cao, M., Dong, Y., Cheung, J. C. K. Hallucinated but Factual! Inspecting the Factuality of Hallucinations in Abstractive Summarization. arXiv Preprint arXiv:2109.09784, 2021. https://doi.org/10.18653/v1/2022.acl-long.236
4. Celikyilmaz, A., Hakkani-Tur, D. Discovery of Topically Coherent Sentences for Extractive Summarization. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, 491-499.
5. Compton, P., Jansen, R. Knowledge in Context: A Strategy for Expert System Maintenance. AI'88: 2nd Australian Joint Artificial Intelligence Conference Adelaide, Australia, November 15-18, 1988 Proceedings 2, Springer Berlin Heidelberg, 1990, 292-306.
6. Devlin, J., Chang, M. W., Lee, K.,Toutanova, K. Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv Preprint arXiv:1810.04805, 2018.
7. Erkan, G., Radev, D. R. LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research, 2004, 22, 457-479. https://doi.org/10.1613/jair.1523
8. Fortunato, S. Community Detection in Graphs. Physics Reports, 2010, 486(3-5), 75-174. https://doi.org/10.1016/j.physrep.2009.11.002
9. Fung, P., Ngai, G., Cheung, C. S. Combining Optimal Clustering and Hidden Markov Models for Extractive Summarization. Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering, 2003, 21-28. https://doi.org/10.3115/1119312.1119315
10. Galgani, F., Compton, P., Hoffmann, A. Citation Based Summarisation of Legal Texts. PRICAI 2012: Trends in Artificial Intelligence: 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, September 3-7, 2012. Proceedings 12, Springer Berlin Heidelberg, 2012, 40-52. https://doi.org/10.1007/978-3-642-32695-0_6
11. Galgani, F., Compton, P., Hoffmann, A. Combining Different Summarization Techniques for Legal Text. Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data, 2012, 115-123.
12. Galgani, F., Compton, P., Hoffmann, A. Summarization Based on Bi-Directional Citation Analysis. Information Processing & Management, 2015, 51(1), 1-24. https://doi.org/10.1016/j.ipm.2014.08.001

13. Ghosh, S., Wyner, A. Identification of Rhetorical Roles of Sentences in Indian Legal Judgments. Legal Knowledge and Information Systems: JURIX, 2019, 3.

14. Grover, C., Hachey, B., Hughson, I. The HOLJ Corpus. Supporting Summarisation of Legal Texts. Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora, 2004, 47-54.

15. Hachey, B., Grover, C. A Rhetorical Status Classifier for Legal Text Summarisation. Text Summarization Branches Out, 2004, 35-42. https://doi.org/10.1145/1165485.1165498

16. Hong, K., Nenkova, A. Improving the Estimation of Word Importance for News Multi-Document Summarization. Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014, 712-721. https://doi.org/10.3115/v1/E14-1075

17. Jain, D., Borah, M. D., Biswas, A. Summarization of Legal Documents: Where Are We Now and the Way Forward. Computer Science Review, 2021, 40, 100388. https://doi.org/10.1016/j.cosrev.2021.100388

18. Landauer, T. K., Foltz, P. W., Laham, D. An Introduction to Latent Semantic Analysis. Discourse Processes, 1998, 25(2-3), 259-284. https://doi.org/10.1080/01638539809545028

19. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension. arXiv Preprint arXiv:1910. 13461, 2019. https://doi.org/10.18653/v1/2020.acl-main.703

20. Lin, C. Y. Rouge: A Package for Automatic Evaluation of Summaries. Text Summarization Branches Out, 2004, 74-81.

21. Lin, H., Ng, V. Abstractive Summarization: A Survey of the State of the Art. Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01), 9815-9822. https://doi.org/10.1609/aaai.v33i01.33019815

22. Liu, Y. Fine-Tune BERT for Extractive Summarization. arXiv Preprint arXiv:1903.10318, 2019.

23. Luhn, H. P. The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, 1958, 2(2), 159-165. https://doi.org/10.1147/rd.22.0159

24. Mallick, C., Das, A. K., Dutta, M., Das, A. K., Sarkar, A. Graph-Based Text Summarization Using Modified TextRank. Soft Computing in Data Analytics: Proceedings of International Conference on SCDA 2018, Springer Singapore, 2019, 137-146. https://doi.org/10.1007/978-981-13-0514-6_14

25. Merchant, K., Pande, Y. NLP Based Latent Semantic Analysis for Legal Text Summarization. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2018, 1803-1807. https://doi.org/10.1109/ICACCI.2018.8554831

26. Mihalcea, R., Tarau, P. TextRank: Bringing Order into Text. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004, 404-411. https://doi.org/10.3115/1220575.1220627

27. Nallapati, R., Zhou, B., dos Santos, C. N., Gulcehre, C., Xiang, B. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. arXiv Preprint arXiv:1602.06023, 2016. https://doi.org/10.18653/v1/K16-1028

28. Narayan, S., Papasarantopoulos, N., Cohen, S. B., Lapata, M. Neural Extractive Summarization with Side Information. arXiv Preprint arXiv:1704.04530, 2017.

29. Newman, M. E. J. Finding and Evaluating Community Structure in Networks. Physical Review E, 2004, 69(26113), 1-16. https://doi.org/10.1103/PhysRevE.69.026113

30. Parveen, D., Ramsl, H. M., Strube, M. Topical Coherence for Graph-Based Extractive Summarization. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, 1949-1954. https://doi.org/10.18653/v1/D15-1226

31. Ren, P., Chen, Z., Ren, Z., Wei, F., Ma, J., de Rijke, M. Leveraging Contextual Sentence Relations for Extractive Summarization Using a Neural Attention Model. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, 95-104. https://doi.org/10.1145/3077136.3080792

32. Rush, A. M., Chopra, S., Weston, J. A Neural Attention Model for Abstractive Sentence Summarization. arXiv Preprint arXiv:1509.00685, 2015. https://doi.org/10.18653/v1/D15-1044

33. See, A., Liu, P. J., Manning, C. D. Get to the Point: Summarization with Pointer-Generator Networks. arXiv Preprint arXiv:1704.04368, 2017. https://doi.org/10.18653/v1/P17-1099

34. Shi, Z., Lin, H., Liu, L., Liu, R., Han, J., Shi, A. Deep Attention Gated Dilated Temporal Convolutional Networks with Intra-Parallel Convolutional Modules for End-to-End Monaural Speech Separation. Interspeech, 2019, 3183-3187. https://doi.org/10.21437/Interspeech.2019-1373

35. Silva, G., Ferreira, R., Lins, R. D., et al. Automatic Text Document Summarization Based on Machine Learn-

ing. Proceedings of the 2015 ACM Symposium on Document Engineering, 2015, 191-194. https://doi.org/10.1145/2682571.2797099

36. Sun, F., Jiang, P., Sun, H., Pei, C., Ou, W., Wang, X. Multi-Source Pointer Network for Product Title Summarization[C]. Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, 7-16. https://doi.org/10.1145/3269206.3271722

37. Vehlow, C., Reinhardt, T., Weiskopf, D. Visualizing Fuzzy Overlapping Communities in Networks. IEEE Transactions on Visualization and Computer Graphics, 2013, 19(12), 2486-2495. https://doi.org/10.1109/TVCG.2013.232

38. Vinyals, O., Fortunato, M., Jaitly, N. Pointer Networks. Advances in Neural Information Processing Systems, 2015, 28.

39. Vodolazova, T., Lloret, E., Muñoz, R., et al. The Role of Statistical and Semantic Features in Single-Document Extractive Summarization. Artificial Intelligence Research, 2013, 2, 35-44. https://doi.org/10.5430/air.v2n3p35

40. Wenbo, W., Yang, G., Heyan, H., Yuxiang, Z. Concept Pointer Network for Abstractive Summarization. arXiv Preprint arXiv:1910.08486, 2019.