ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230553

Towards Legal Judgment Summarization: A Structure-Enhanced Approach

Qiqi Wang[a], Ruofan Wang[a], Kaiqi Zhao[a], Robert Amor[a], Benjamin Liu[a], Xianda Zheng[a], Zeyu Zhang[b,a] and Zijian Huang[a]

[a]The University of Auckland
[b]Huazhong Agricultural University
{qwan857, rwan551, xzhe162, zzha669, zhua764}@aucklanduni.ac.nz, {kaiqi.zhao, r.amor, b.liu}@auckland.ac.nz

Abstract. Judgment summaries are beneficial for legal practitioners to comprehend and retrieve case law efficiently. Unlike summaries in general domains, e.g., news, judgment summaries often require a clear structure. Such a structure helps readers grasp the information contained in the summary and reduces information loss. To the best of our knowledge, none of the existing text summarizers can generate summaries aligned with the summary structure in the legal domain. Inspired by this observation, this paper introduces a Summary Structure-Enhanced (SSE) method to synthesize structured summaries for legal documents. SSE can easily be incorporated into the Encoder-Decoder framework, which is commonly adopted in state-of-the-art text summarizers. Experiments on the datasets of New Zealand and Chinese judgments show that the proposed method consistently improves the performance of state-of-the-art summarizers in terms of Rouge scores.

1 Introduction

In the legal domain, judgment summaries help legal experts quickly understand the case information and search for case law to support their claims. Meanwhile, text summarization models have been developed to automate the generation of summaries [18, 28, 23]. Existing text summarization methods are typically categorized into *extractive* and *abstractive* methods. Extractive methods summarize a document by *selecting* several representative sentences from the document. Abstractive methods process the source document and *generate* a concise text to summarize the document.

When applied to legal judgments, existing text summarizers ignore the structure features from summaries. It leads to generated summary information loss. Specifically, they concentrate too much on specific aspects in judgments, such as facts, while ignoring the others, like decision reasons. For instance, Figure 1 shows a legal judgment summary written by a legal expert versus the summaries generated by the state-of-the-art extractive [27] and abstractive [13] methods. The expert-written summary follows a clear structure that contains three key parts, namely *Decision*, *Facts*, and *Reasons*. We can observe that neither the extractive summary nor the abstractive summary is close to the human-generated summary. Specifically, the extractive summary is incomplete and excessive – it does not contain the crucial information relating to the alleged breach of law in the *Facts* component, and it has no *Reasons* component while offering too much information in the *Decision* component. Similarly, the abstractive summary does not include the *Facts* component and the key information on *Reasons*. We believe that the lack of a summary structure in the existing models is the main reason for information loss in the resulting summaries.

HUMAN: successful application by d and t for approval of a settlement of their privacy breach claim. [ccll have send private information regard the two child of d and m to a third party and fail to mitigate the harm cause. the privacy commissioner have find a breach of privacy. d and m have claim compensation for themselves and the child , who be minor at that time. settlement amount agree with ccll be (a) $ 80,000 for each child , (b) $ 25,000 each for d and m and (c) $ 8,000 towards cost. payment to child be to be hold in trust until each turn 30.] {held , settlement be appropriate. deed of declaration of trust for the child be appropriate}

Extractive: [The applicants seek the court's approval of a settlement of a claim which their children have for compensation. The applicants, d and t, are the parents of g and a. t and d were in a de facto relationship from july 2001. they married in 2005 but separated in august 2007. their children reside primarily with t.] and (c) there is no order as to costs and disbursements. [the respondent is a law firm based in christchurch.] (b) leave is reserved to counsel to seek further directions should such be necessary. [a solicitor within the firm acted as counsel for g and a in family court matters between 2014 and 2018.] I order: "(a) the proposed settlement agreement between the applicants and the respondents dated 16 march 2021 is approved.

ABSTRACTIVE: unsuccessful application by d and t for approval of settlement of claim which their child have for compensationd and t seek approval of a settlement of a claim which they have for personal benefit. {hold, agreement make a sought}

Figure 1. Example summaries produced by human, the state-of-the-art extractive summarizer ExtSum-LG [27] and abstractive summarizer BART [13]. Sentences annotated by __, [ ], and { } represent the *Decision*, *Facts*, and *Reasons* components, respectively.

We argue that summary structure is important for generating high-quality summaries in the legal domain. Some recent works in the general domain noticed that structured information is helpful for summarization, but they focus on either the source document structure [27, 15] or sentence relations [1]. Yet, summaries generated using these approaches do not follow any summary structure. Gidiotis et al. [9] propose a one-to-one mapping between the document sections with summary sections and generates each summary section from the corresponding document section. However, legal judgments vary case by case and judge by judge, with no explicit structure that aligns with the expert-generated summaries. In the legal domain Elaraby et al. [6] propose a supervised learning method called Ar-

gLegalSumm to mine the argumentative structure of the source document and classify the sentences into three predefined groups for judgments of various countries: *Issues*, *Reasons* and *Conclusions*. Experts often follow particular summary structures, which may vary across jurisdictions (See Section 3.1). With a predefined structure, ArgLegalSumm cannot cater to the various requirements of the summary structures in different regions. Moreover, the strong assumption that one sentence must belong to one of these parts in legal judgments is difficult to establish [3, 2]. For instance, comments on the facts may relate to both *Facts* and *Reasons*.

To the best of our knowledge, the existing structured summarizers all have strong assumptions on the source document structure, which do not hold in the legal domain. This work aims to propose a new approach to generating summaries by following a given *summary structure* without specific assumptions on the source document structure. One straightforward solution is to train a separate summarizer for each part of the summary structure. Each summarizer takes the whole source document as input and generates the summary for a specific part of the summary, e.g., facts. However, using a summarizer for each part is costly in terms of memory and training time. Furthermore, this straightforward solution, which separates out each part summarizer, discards the relations among different parts.

Based on these, we design a novel Summary Structure-Enhanced (SSE) method that can be incorporated into the Encoder-Decoder framework for generating structured summaries for judgments. With a pre-defined summary structure containing several key parts (e.g., *Decision*) for a specific jurisdiction's requirement, SSE maintains a latent space for each summary part and calculates a document-specific part representation for each summary part by projecting the document representation (output from the encoder) into various latent spaces, each corresponding to a summary part. The projected representations indicate the content to be covered in each part. Then, we use the document-specific part representation to align words or sentences to the summary parts and compute a word-specific or sentence-specific part representation, which is passed to the decoder for producing the summary. In this way, the resulting summary covers the information of different parts to produce a comprehensive and balanced summary. The proposed SSE method has two salient features: (1) It can be integrated into the existing state-of-the-art text summarizers with the Encoder-Decoder framework; (2) A single summarizer with SSE can output summaries for all summary parts at the same time, thus saving memory and training time compared to the straightforward solution. Our main contributions include:

- This work is the first to study the structured summary generation problem on legal judgments with no specific document structure. We propose a straightforward solution to solve the problem.
- To reduce the memory usage and training time cost, we further propose a novel Summary Structure-Enhanced (SSE) method to align the information of the source document with different summary parts. SSE can be applied to existing summarizers with the Encoder-Decoder framework.
- We evaluate the proposed methods on two judgment datasets from different countries. Results show that the proposed methods can improve the summarization quality.

2 Related Work

2.1 Extractive Summarization

Extractive summarization selects words or sentences that capture the most important content in a document to form a summary. SummaRuNNer [21], one of the earliest methods to utilize an Encoder-Decoder structure, adopts two GRU-RNN layers as the encoder to obtain a vector representation of each sentence. A classifier is adopted as the decoder to decide which sentence to be selected as part of the summary using the sentence representations. ExtSum-LG [27] extends the sentence-level encoder of SummaRuNNer to obtain multi-level representations, including sentence, document, and section representations. The following research change the RNN Encoder to a Transformer encoder [24, 29] or BERT encoder [25, 17]. In addition, some research adopt separate encoders to obtain the sentence-level and document-level representations [17, 29], while some work use randomly initialized trainable sentence and document representations [24]. Furthermore, Memsum [11] introduces a reinforcement learning method for considering the selection history to reduce the redundancy of generated summaries.

2.2 Abstractive Summarization

Abstractive summarization generates a summary word-by-word. The Encoder-Decoder framework is mostly adopted in this task. The decoder has a similar structure to the encoder. BERT [5] and BART [22], two famous language models, utilize the Transformer Encoder-Decoder structure to obtain summaries. Based on the structure, previous research introduce Entity Aggregation [10], Key Phrases Detection [14], Sentence Structure Relations [1], and Time Content Selection [4] to generate summaries.

2.3 Legal Document Summarization

Legal documents are different from many other types of documents because they are typically lengthy and require specific domain knowledge to understand. Legal knowledge, legal documents cited history [7], legal sentence syntactic knowledge [12], etc., are used in extractive summarizers to improve the relatedness of selected sentences. Most existing abstractive summarizers apply pre-trained language models, which have a length restriction. Since legal documents are naturally longer, Se3 [20] cuts the document into several chunks to improve the quality of summaries. ArgLegalSumm [6] introduces the argumentative structure of documents and groups sentences into different arguments using a classifier. The argument label and representation of each sentence are then passed to an Encoder-Decoder model for generating the summary.

To summarize, the Encoder-Decoder framework is the most common model architecture among the existing works in text summarization. However, previous works ignore the summary structure adopted by human experts when they summarize documents.

Table 1. Summary structure for New Zealand judgments

Part	Description	Example
Decision	What the court decides	unsuccessful application by r for leave to commence proceeding.
Fact	Essential factual information of the case	r a vexatious litigant; r seeking to commence proceeding seek 'exemplary damage ' of ' $ 900 trillion ' against the high court for allegedly fail to process an application for leave to appeal a judgment from 2013.
Reason	Reason for the decision	Held, proposed proceeding frivolous and vexatious.

Table 2. Summary structure for Chinese judgments

Part	Detail	Sample
Type	Category of case	The case is a dispute over a loan contract.
Plaintiff's Claims	Claims from plaintiff(s)	The plaintiff filed a claim for return of the principal amount of a loan together with normal interest and penalty interest.
Defendant's Response	Response from defendant(s)	The defendant admitted that the loan was genuine and agreed to return it.
Reasons	Why make the decision	The court found: 1. there existed a the loan contractual relationship; 2. the defendant was late in returning the loan.
Decision	What the court decides	According to Section 206 and 207 of the Contract Law, the defendant is to return the principal and pay interest on the loan.

3 Structured Summarization

3.1 Motivation

Referring to the example in Figure 1, we observe that a summary of a New Zealand judgment often contains three main components: (1) **Decision** that describes the case outcome; (2) **Facts** between litigation participants, which are the essential information of the relevant cases; (3) **Reasons** that explain why the court made the relevant decision. Table 1 shows the summary structure for New Zealand judgments.

We extend our observation to the Chinese judgment dataset, 2020CAIL-SFZY [8]. Similarly, there is a summary structure with five parts: (1) **Type** of the case; (2) **Plaintiff's Claims**; (3) **Defendant's Response**; (4) **Reasons**, and (5) **Decision**, as illustrated in Table 2.

We observe that different jurisdictions' judgment summaries have different structures because the legal litigants may focus on different information. For instance, the **Type** part of summaries is essential for Chinese legal experts to quickly understand, within over 400 categories under the Chinese civil law, which category the relevant case falls into [26]. Although different jurisdictions focus on different information in the law leading to summaries differences, clear structures can always be observed.

Motivated by the observations above, we propose the research problem of generating structured summaries by following a given summary structure. In the next subsections, we first introduce the Encoder-Decoder framework used in most recent summarizers (Section 3.2). Then, we discuss a straightforward solution of utilizing existing summarizers to generate structured summaries and its limitations in memory and training time cost and learning cross-part relations (Section 3.3). In Section 3.4, we elaborate on the proposed **Summary Structure Enhanced (SSE)** method to leverage the summary part information and reduce the memory requirement and training time.

3.2 Text Summarizers with the Encoder-Decoder Framework

As discussed in Section 2, most recent text summarizers adopt the Encoder-Decoder framework. The top of Figure 2 shows the architecture of a vanilla summarizer. Under both extractive methods and abstractive methods, the encoder has the same purpose of understanding source documents. Specifically, the encoder takes the document, which is represented by a sequence of tokens $\{t_1,...,t_n\}$, as input and outputs a vector representation h_{t_i} for each token t_i. For extractive methods, sentences or words can be regarded as input tokens, while for abstractive methods, words are typically regarded as input tokens. The token representations can be used in a decoder for producing summaries.

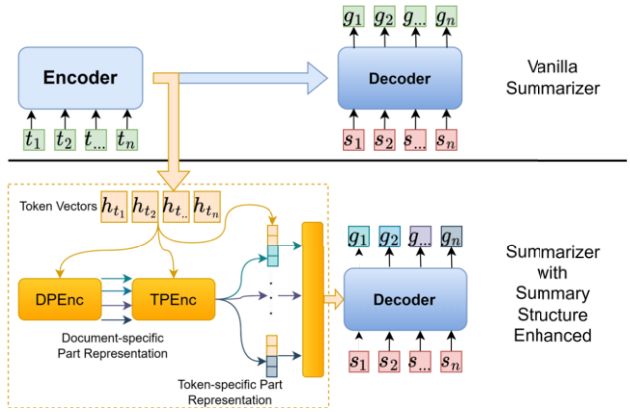

Figure 2. Summarizer architectures. The top is a vanilla summarizer with the Encoder-Decoder framework. The bottom is the proposed Encoder-SSE-Decoder framework that incorporates the proposed SSE block.

For extractive methods, the decoder is a classifier that outputs the probability g_i of selecting each sentence (or word) [27]. For abstractive methods, the decoder is usually a sequence model with RNNs or a Transformer, and it can generate each word g_i sequentially. For generalization purposes, we use the same notation g_i to denote the i-th output of the decoder for both extractive or abstractive summarizers.

3.3 Independent part summarizer

Next, we proceed to present a straightforward method for the structured summarization problem using existing summarizers. We name this method Independent Part Summarizer (IPS). This method trains several independent sub-summarizers for the same structure. The input to all sub-summarizers is the same sequence of tokens $\{t_1,...,t_n\}$ of a given document. Each sub-summarizer is responsible for generating a summary for each summary part. The outputs of the sub-summarizers are concatenated to obtain the complete summary.

While IPS is a straightforward method to incorporate the structure summary information, it has two shortcomings: (1) Since IPS needs to train a separate summarizer for each summary part, the memory consumption and training time increase with the number of parts; (2) Each sentence or word in a judgment may be related to multiple summary parts [3, 2]. The sub-summarizers are optimized independently and thus may select or generate the same words and sentences for different parts, resulting in redundancy.

3.4 Summary Structure Enhanced Summarizer

To address the shortcomings of IPS, we propose a Summary Structure Enhanced (SSE) method. The proposed SSE method can be applied to existing models with the Encoder-Decoder framework. Summarizers with SSE not only utilize the information of a given sum-

mary structure but also keep memory size and training time as small as possible.

3.4.1 Overall Architecture

The bottom of Figure 2 shows the architecture of the summarizer with SSE. Compared to the Encoder-Decoder structure (the top part of Figure 2), SSE can be inserted between the encoder and decoder. Compared to the IPS summarizer, the SSE method only needs one encoder and one decoder. The Summary Structure Enhanced (SSE) block contains three modules: (1) Document-specific Part Encoding (DPEnc); (2) Token-specific Part Encoding (TPEnc); (3) Fusion layer. Given a document and a pre-defined summary structure, SSE will utilize the DPEnc module to learn a latent representation for each summary part, which implies the content that the summary part should cover. Then, the TPEnc module aligns the tokens to each part according to the similarity between the token and part representations. In this way, tokens that are unrelated to a specific summary part are less likely to be selected. The fusion layer is to combine the token representation and token-specific part representation obtained from the encoder and the TPEnc module, respectively. The fused representations of the tokens are then passed to the decoder for generating part-specific summaries.

3.4.2 Document-specific Part Encoding (DPEnc)

Since the content of each summary part depends on the document to be summarized, the DPEnc module computes part representations from the document representation. Specifically, given the encoder output, i.e., the token encodings (h_{t_1}, h_{t_2}, ..., h_{t_n}), the DPEnc module takes two steps. In the first step, it applies Multi-head Pooling (MHP) [16] to learn the document representation h_d from the encoder output, which captures the content information of the document. In the second step, it defines a latent space for each part and projects the document representation to different part spaces. The projected representation encodes the content information for a specific part and thus is used to represent the content information to be covered in part. The projected representation of the j-th summary part for a document d is:

$$p_d^j = \mathbf{W}_p^j h_d, \tag{1}$$

where $\mathbf{W}_p^j$ is a learnable weight matrix for the j-th part, which projects the document representation to the j-th summary part.

3.4.3 Token-specific Part Encoding (TPEnc)

With the document-specific part representation, the TPEnc module aligns the tokens in the source document to each summary part. Unlike ArgLegalSumm [6], we do not assume each sentence must belong to only one summary part. This is because a sentence or word in a judgment may contain information that belongs to several parts [3, 2].

Instead, we propose to compute part representations for each token, which encodes the information related to a specific summary part. This design allows each token to be aligned with multiple parts. Specifically, we calculate the similarity between the token representation and the document-specific representation of each part. Then we obtain the token-specific part representation $\hat{p}_{t_i}^j$ by multiplying the document-specific part representation of the j-th part with the similarity between the token:

$$\begin{aligned} \hat{p}_{t_i}^j &= {a_{t_i}}^j p_d^j \\ {a_{t_i}}^j &= \text{Softmax}(h_{t_i} {p_d^j}^T), \end{aligned} \tag{2}$$

where ${a_{t_i}}^j$ is the similarity between given sentences (or words) and j-th part.

3.4.4 Fusion layer

The token-specific part representation only encodes the information of summary parts related to the token without capturing the sentence's content. The fusion layer is to combine the content information and part information of a token. Concretely, the fusion layer concatenates the token encoding h_{t_i} and the summary part-related representation $\hat{p}_{t_i}^j$, and then applies a linear transformation:

$$p_{t_i}^j = \mathbf{W}_t [h_{t_i} \| \hat{p}_{t_i}^j]. \tag{3}$$

In Equation 3, $\mathbf{W}_t$ represents a learnable weight matrix and $[\cdot\|\cdot]$ is the concatenation operation. The concatenation operation can be replaced by other aggregation functions, such as multi-layer perceptrons. For simplicity, we use concatenation in this paper and leave the study of different aggregation functions as future work. The output of the Fusion layer $p_{t_i}^j$ is the representation of token t_i related to the j-th summary part, which is then fed into the decoder for generating a summary for each part. Similar to IPS, we concatenate the summaries for all parts to obtain the complete summary.

3.4.5 Extensions

The proposed SSE block can be easily extended to support additional information such as document representations [27], selection history representations [11], sentence structure representations [1], etc., which are used in some existing summarizers. To achieve this, one can concatenate the additional information with the content representation h_{t_i} and token-specific part representation $\hat{p}_{t_i}^j$ in the fusion layer defined in Equation 3.

3.5 Training

The proposed SSE method can be adopted in both extractive and abstractive methods. The parameters in SSE are trained by comparing the generated summaries and ground-truth (or reference) summaries. We present the loss functions of two types of summarizers in this subsection.

Extractive method An extractive summarizer selects some sentences from the source document as a summary. Therefore, the number of selected sentences is much smaller than the number of sentences in the source document. It leads to an imbalance issue which means there are many more negative labels than positive labels [16]. Following previous work, we design the weighted negative log-likelihood for different parts as, $w^j = \frac{\#positive^j}{\#negative^j}$. The loss function is:

$$\begin{aligned} \mathscr{L} = \sum_{j \in m}^{m} - \sum_{(d,y_i) \in \mathscr{D}_{train}} & (w^j * y_i \log p(y_i | g_i^j)) \\ & + (1 - y_i) \log p(y_i | g_i^j)), \end{aligned} \tag{4}$$

where g_i^j is the probability of selecting the i-th sentence for the j-th summary part, and $(d, y_i) \in \mathscr{D}_{train}$ denotes a training pair of document d and the label y_i of selecting the i-th sentence for the summary. If the i-th sentence is selected in the reference summary, then $y_i = 1$, otherwise, $y_i = 0$.

Abstractive method We follow previous work [13] and extend its loss function for abstractive summarization to our problem:

$$\mathscr{L} = \sum_{j \in m}^{m} - \sum_{(d, z_i) \in \mathscr{D}_{train}} (z_i \log p(z_i | g_i^j)), \tag{5}$$

where $(d, z_i) \in \mathscr{D}_{train}$ is a training pair of document d and its summary token label z_i.

4 Experiments

We evaluate the effectiveness of the SSE method by addressing the following research questions:

RQ1. How much is the performance of generated summaries improved with structure summary?

RQ2. What are the memory and training time requirements of the SSE method compared to IPS?

RQ3. How does the SSE method avoid the generated summary missing information and information imbalance issues?

RQ4. What are the impacts of the modules of the SSE method on the summarization performance?

Table 3. Statistics of New Zealand Judgment Dataset and CAIL2020-SFZY. #, avg., doc., summ., and Sent. represent the Number of, average, document, summary, and sentences, respectively.

Dataset	#avg. doc.		#avg. summ.	
	#Words	#Sent.	#Words	#Sent.
NZJD	2819.5	121.4	209.4	11.3
-Decision	-	-	20.2	1.1
-Fact	-	-	102.6	5.2
-Reason	-	-	86.6	5.0
CAIL2020-SFZY	2629.5	57.4	283.7	7.3
-Type	-	-	10.9	1.0
-Plaintiff Claim	-	-	36.4	2.1
-Defendant Answer	-	-	33.7	1.3
-Reason	-	-	106.5	2.0
-Decision	-	-	88.7	1.3

Table 4. Training time (in minutes) of different models

Model	Dataset	ExtSum-LG	Memsum
Vanilla	NZJD	34	206
	CAIL.	52	136
IPS	NZJD	149	754
	CAIL.	195	576
SSE	NZJD	251	545
	CAIL.	149	329

4.1 Dataset

CAIL2020-SFZY was published by China AI & Law Challenge and contains several types of Chinese judgment documents[1]. Summaries for Chinese judgments are well-structured texts. It is easy to capture the 'Type' part by selecting the first sentence of the summary. And 'plaintiff' and 'defendant' are two guide signals to mine the 'Plaintiff Claims' and 'Defendant Answers' components. The remaining parts can be divided by a fixed term, 'The court decides'. In the Rouge evaluation for Chinese sentences, we compare it based on the words instead of the characters. We utilize the Chinese Legal Language-based word segmentation tool, Lawa[2], to partition sentences into words.

Moreover, we constructed a new dataset, New Zealand judgment dataset (NZJD)[3], for this experiment. NZJD contains 6,155 judgments from all levels of courts in New Zealand. We split the summary into three parts by the following process: (1) we extract the first sentence from the summary as the "Decision." Since every summary in the New Zealand Judgments utilizes a single sentence to summarize the decision. (2) we identify the keyword "held" and use it to divide the remaining text into "Facts" and "Reason".

Table 3 shows the statistics of the two datasets. We split each summary of the two datasets into several parts by following the observation in Section 3.1. For each summary part, we set the maximum length of the generated text as the average length of the part in reference summaries. We utilize the same oracle methods as the base models to construct extractive summarization labels for the two datasets.

Table 5. Memory size of different models. The memory size of ExtSum-LG is in MB. Others are in GB.

Model	Dataset	ExtSum-LG	Memsum	BART
Vanilla	NZJD	2.30	0.38	0.54
	CAIL.	2.76	1.56	0.54
IPS	NZJD	6.90	1.01	1.56
	CAIL.	13.80	7.80	2.61
SSE	NZJD	5.98	0.40	0.55
	CAIL.	7.66	1.61	0.56

4.2 Compared Models and Configurations

Extractive Methods We show the flexibility of SSE and its effectiveness by applying it to several representative summarizers. ExtSum-LG [27] proposed the general structure of an extractive summarizer. As discussed in Sec 2, many state-of-the-art works [24, 25, 29, 19] follow its design in general. Therefore, we select ExtSum-LG as one of the baseline models to apply our SSE method. Furthermore, we also utilize the state-of-the-art summarizer Memsum [11], which is based on reinforcement learning. We train all baseline models by following the settings reported in their papers.

Abstractive Methods As the pre-trained language model is the most popular model used in abstractive summarizers, we apply the SSE method to BART [13]. We utilize the HuggingFace pre-trained BART[4] and follow the same settings of Se3 [20], a summarizer designed for the legal domain.

We keep all the settings of the backbone model when training the proposed SSE method. Besides, we set the head of MHP as 8. And all learnable parameters, those not included in the backbone settings, are initialized by the standard normal distribution. All experiments run on an Nvidia GeForce RTX 3090 GPU platform with 24GB memory.

[1] CAIL2020-SFZY dataset: https://github.com/china-ai-law-challenge/CAIL2020/tree/master/sfzy.

[2] https://pypi.org/project/lawa/

[3] The pre-processed NZJD dataset will be released in https://github.com/77-qiqi-wang/SSE.

[4] https://huggingface.co/facebook/bart-base

Table 6. Result on New Zealand Judgment Dataset and CAIL2020-SFZY. We did the paired t-test between SSE and IPS on overall Rouge scores. The p-values are all less than 0.01. The improvement of SSE over IPS is statistically significant.

Model	Dataset	ExtSum-LG	Memsum	BART
		R1/R2/RL	R1/R2/RL	R1/R2/RL
Vanilla	NZJD	33.78 / 12.15 / 29.47	39.87 / 14.48 / 36.47	38.35 / 17.74 / 34.45
	CAIL2020-SFZY	43.97 / 25.09 / 39.05	34.57 / 17.60 / 29.76	43.62 / 24.23 / 37.07
		R1/R2/RL	R1/R2/RL	R1/R2/RL
	NZJD	35.37 / 13.36 / 30.32	40.70 / 15.91 / 37.56	39.93 / 19.20 / 37.28
	-Decision	28.26 / 12.14 / 25.15	31.98 / 14.12 / 28.50	63.27 / 48.85 / 61.99
	-Fact	29.94 / 11.33 / 23.01	33.24 / 13.17 / 29.99	32.53 / 13.44 / 28.01
	-Reason	27.16 / 9.41 / 20.86	31.44 / 11.80 / 28.41	25.37 / 9.98 / 23.31
IPS	CAIL2020-SFZY	53.64 / 35.40 / 49.40	50.24 / 31.17 / 45.77	48.46 / 28.50 / 42.66
	-Type	19.68 / 8.52 / 19.61	31.27 / 17.76 / 31.24	80.37 / 72.70 / 80.37
	-Plaintiff Claim	44.99 / 26.95 / 43.99	47.90 / 29.90 / 46.82	65.16 / 46.35 / 64.17
	-Defendant Answer	43.21 / 23.21 / 41.65	45.09 / 23.98 / 43.53	50.19 / 31.30 / 48.85
	-Reason	52.16 / 37.25 / 48.99	48.56 / 33.79 / 45.42	32.43 / 12.55 / 26.57
	-Decision	51.65 / 39.99 / 50.25	39.72 / 23.85 / 37.70	49.68 / 26.73 / 46.49
		R1/R2/RL	R1/R2/RL	R1/R2/RL
	NZJD	**37.04 / 14.18 / 31.91**	**41.63 / 16.58 / 39.07**	**44.13 / 22.01 / 41.25**
	-Decision	29.03 / 12.60 / 25.85	32.88 / 14.51 / 29.38	66.31 / 51.26 / 64.47
	-Fact	30.72 / 11.43 / 23.42	35.24 / 14.62 / 31.93	36.47 / 16.31 / 31.52
	-Reason	28.23 / 9.83 / 21.63	32.43 / 12.52 / 29.47	29.28 / 11.40 / 26.37
SSE	CAIL2020-SFZY	**54.19 / 35.98 / 49.94**	**51.21 / 31.75 / 46.59**	**51.45 / 31.53 / 45.75**
	-Type	19.38 / 8.33 / 19.32	31.21 / 17.77 / 31.18	83.51 / 77.76 / 83.51
	-Plaintiff Claim	45.33 / 27.36 / 44.40	48.01 / 30.11 / 46.84	69.29 / 51.66 / 68.12
	-Defendant Answer	43.93 / 24.09 / 42.37	45.17 / 24.03 / 43.50	52.15 / 32.57 / 50.74
	-Reason	52.61 / 37.93 / 49.51	50.63 / 35.04 / 47.39	36.08 / 15.03 / 30.17
	-Decision	52.03 / 40.46 / 50.68	39.96 / 24.16 / 37.91	51.92 / 30.09 / 48.59

Table 7. Ablation studies on New Zealand Judgment Dataset with ExtSum-LG model. We did the two-sample t-test between the proposed SSE and other methods on overall scores. ♠ means the p-value is less than 0.05. Otherwise, the p-value is less than 0.01.

	Overall	Part 1	Part 2	Part 3
SSE	37.04 / 14.18 / 31.91	29.03 / 12.60 / 25.86	30.72 / 11.43 / 23.42	28.23 / 9.83 / 21.63
SSE w/o DPEnc	35.93 / 13.76♠ / 31.17♠	30.02 / 13.13 / 26.69	29.67 / 10.98 / 22.68	26.92 / 9.18 / 20.72
SSE w/o TPEnc	36.00 / 13.90♠ / 31.13♠	29.29 / 12.71 / 26.15	29.61 / 11.01 / 22.64	27.43 / 9.47 / 21.02
SSE w/o DPEnc & TPEnc	35.91 / 13.71♠ / 31.10♠	29.03 / 12.20 / 25.80	29.73 / 10.98 / 22.64	26.83 / 9.01 / 20.67

4.3 Result and Discussion

RQ1: Comparison of Summarizers with or without Summary Structure. Table 6 reports the Rouge-1, Rouge-2, and Rouge-L of all baseline summarizers with and without using IPS and SSE. We can summarize three key findings from the results. First, IPS and SSE both improve the quality of generated summaries compared to the vanilla summarizer. This proves that the summary structure is essential for summarization. Second, SSE consistently outperforms IPS in terms of overall Rouge scores because SSE aligns the sentences or words of the source document to different parts and optimizes the part representations simultaneously. In this way, information is shared across different parts, and thus SSE can reduce redundant and unrelated information. Third, SSE outperforms IPS in generating summaries for each part in most cases. Especially, SSE significantly improves the performance of longer parts, such as *Facts* and *Reasons*. This is because SSE considers different parts in a unified way, and thus it can select or generate more interrelated part summaries.

RQ2: Memory size and training time of SSE. As discussed in Section 3.3, SSE addresses the limitations of IPS in terms of memory usage and training time. Table 5 and Table 4 report the memory usage and training time for the vanilla models, IPS and SSE, respectively. We found that SSE requires less memory and training time compared to IPS. This is because SSE no longer trains multiple summarizers to generate structured summaries. A larger reduction of memory usage can be observed on CAIL2020-SFZY, because it has more summary parts. Nonetheless, there is little improvement in lightweight models like ExtSum-LG. ExtSum-LG only contains bi-RNN and MLP layers, both of small sizes. Compared to the vanilla models, its SSE counterparts do not incur much extra memory usage, especially when the vanilla models are large. Similarly, SSE saves training time because it trains a single summarizer instead of multiple summarizers, as does IPS.

RQ3: Case Study. Figure 3 shows the summary generated by using summaizers, ExtSum-LG [27] for extractive and BART [13] for abstractive summaries, with the SSE method for the same source document of the examples illustrated in Figure 1.

After applying the SSE method, the extractive summary now includes the *Reasons* part (in braces). Moreover, the *Facts* (in brackets) in the extractive summary now include the crucial information relating to the alleged breach of law. Also, the extractive summary generated without SSE contains out-of-order sentences, which makes the summary confusing to readers and, as a result, greatly diminishes its usefulness. In contrast, the sentences in the extractive summary generated with SSE do follow a proper order. As for the abstractive summary, the improvements are even more impressive. The summary produced with the SSE method now contains all three parts: *Decision*, *Facts*, and *Reasons*. Furthermore, unlike the summary produced without the SSE method, the *Reasons* part is now a full sentence and

HUMAN: successful application by d and t for approval of a settlement of their privacy breach claim. [ccll have send private information regard the two child of d and m to a third party and fail to mitigate the harm cause. the privacy commissioner have find a breach of privacy. d and m have claim compensation for themselves and the child , who be minor at that time. settlement amount agree with ccll be (a) $ 80,000 for each child , (b) $ 25,000 each for d and m and (c) $ 8,000 towards cost. payment to child be to be hold in trust until each turn 30.] {held , settlement be appropriate. deed of declaration of trust for the child be appropriate}

Extractive w/ SSE: Accordingly I will make an order under s 104 Contract and Commercial Law Act 2017 approving the proposed settlement with CC. [The applicants seek the Court's approval of a settlement of a claim which their children have for compensation. t and d were in a de facto relationship from July 2001. On 28 August 2020 the applicants on their own behalf and on behalf of their children reached a settlement in principle of the claim. During the period the solicitor acted for the children, it is alleged the solicitor sent eleven emails containing private information relating to d and t and their children to an incorrect email address. The applicants, on behalf of themselves and their children, made a claim for compensation against the firm.] {Approval of the settlement requires me to be satisfied that the terms of the Trusts are appropriate. The deeds of declaration contain comprehensive, standard provisions. To achieve the settlement reached without the cost of a hearing and receiving a contribution to costs represents a reasonable outcome. I am satisfied they are appropriate.}

ABSTRACTIVE w/ SSE: successful application by d and t for approval of a settlement agreement between them and their child. [d and t have make claim for compensation against ccll. solicitor act for d and others in family court matter between 2014 and 2018. d have file a complaint with the office of privacy commissioner ('opp') in relation to disclosure of private information relate to d and her child. d and t have lodge a complaint to the opp in relation. d apply for approval of the proposed settlement.] {hold, satisfy that the propose settlement be in line with the interest of everyone party to the settlement.}

Figure 3. Summaries produced by summarizers with SSE for the same document used in Figure 1. Sentences annotated by __, [ ], and { } represent the *Decision*, *Facts*, and *Reasons* components, respectively.

HUMAN: successful application by d and t for approval of a settlement of their privacy breach claim. [ccll have send private information regard the two child of d and m to a third party and fail to mitigate the harm cause. the privacy commissioner have find a breach of privacy. d and m have claim compensation for themselves and the child , who be minor at that time. settlement amount agree with ccll be (a) $ 80,000 for each child , (b) $ 25,000 each for d and m and (c) $ 8,000 towards cost. payment to child be to be hold in trust until each turn 30.] {held , settlement be appropriate. deed of declaration of trust for the child be appropriate}

Extractive w/ IPS: The applicants seek the Court's approval of a settlement of a claim which their children have for compensation. [he applicants seek the Court's approval of a settlement of a claim which their children have for compensation. t and d were in a de facto relationship from July 2001. On 28 August 2020 the applicants on their own behalf and on behalf of their children reached a settlement in principle of the claim. The applicants, on behalf of themselves and their children, made a claim for compensation against the firm. During the period the solicitor acted for the children, it is alleged the solicitor sent eleven emails containing private information relating to d and t and their children to an incorrect email address.] {I am satisfied they are appropriate. To achieve the settlement reached without the cost of a hearing and receiving a contribution to costs represents a reasonable outcome. Approval of the settlement requires me to be satisfied that the terms of the Trusts are appropriate. The deeds of declaration contain comprehensive, standard provisions.}

ABSTRACTIVE w/ IPS: successful application by d and t for approval of settlement of claim which their child have for compensation. [d and t have be in a de facto relationship from 2004 to 2007. d and t file complaint with the privacy commissioner in relation to the disclosure of private information. d now over 18 year of age and t consent to the order sought.] {hold, d and t fit and proper person to enter into the agreement on behalf of the minor}

Figure 4. Summaries produced by summarizers with IPS for the same document used in Figure 1. Sentences annotated by __, [ ], and { } represent the *Decision*, *Facts*, and *Reasons* components, respectively.

contains meaningful information. These improvements verify that the SSE method is effective.

We also conduct the same case study for IPS. Figure 4 shows the IPS method generates summaries by using the same backbone summarizers and the same source document used in Figure 1 and Figure 3. After applying the IPS method, the extractive summary now includes the *Reasons* part (in braces), compared to the vanilla summarizers (Figure 1). Moreover, the *Facts* (in brackets) in the extractive summary now contain crucial information related to the alleged breach of law. However, the *Decision* part (underlined) is incorrect. The sentence describes the fact, which is similar to the first sentence in *Facts* part. Compared to the SSE method output (Figure 3), it proves considering different parts together can select more interrelated part summaries. As for the abstractive summary, IPS also improves the vanilla summarizers (Figure 1). The summary produced by the IPS method now contains all three parts: *Decision*, *Facts*, and *Reasons*. Furthermore, unlike the vanilla summarizer, the *Decision* part is correct and factual. However, some decision reasons remain absent in the *Reasons* part. The result verifies that the SSE method is more effective than IPS.

RQ4: Ablation study. We make three ablations to SSE to verify the effectiveness of the key components of SSE: (1) SSE without DPEnc, which replaces the document-specific part encodings by randomly initialized encodings; (2) SSE without TPEnc, which removes the token-specific encoding and pass the results from DPEnc directly to the Fusion layer; (3) SSE without DPEnc and TPEnc, which removes both DPEnc and TPEnc by directly applying different decoders to generate summaries for different parts.

The results are reported in Table 7. We can observe that the complete SSE consistently outperforms all its ablations. Removing any of the key components results in a significant performance drop. The ablation study verifies that both DPEnc and TPEnc are indispensable. When removing DPEnc, the proposed method performs better on the *Decision* part. This is because after removing DPEnc, we initialize part representation randomly and use the same part representations for all documents. The shared part representations better fit the *Decision* part as the *Decision* content is often similar across different documents. In contrast, for summary parts that exhibit high variance across documents, e.g., *Facts* and *Reasons*, removing DPEnc causes significant performance degradation. The results justify DPEnc's effectiveness in producing part representations that are specific to the source document's content. Considering the overall performance, DPEnc is effective and indispensable in most cases.

5 Conclusion

This paper finds summaries in the legal domain are well-structured and proposes to use the structure information to improve the generated summary quality. First, we present a straightforward solution to apply existing summarizers to the structured summary. We named the method Independent Part Summarizer, IPS. It can generate each part summary by training based on each specific part summary. But the approach costs a lot of memory and training time. Then, we propose a Summary Structure Enhanced (SSE) method to align the content of the document to different summary parts and generate part-specific summaries. SSE can be applied to state-of-the-art summarizers with the Encoder-Decoder framework. SSE does not need to train several times for different parts. Therefore, SSE is more efficient than IPS. Experiments on two legal document datasets from two countries show that SSE improves the quality of generated summaries.[5]

[5] The code and datasets used in this paper will be released in https://github.com/77-qiqi-wang/SSE.

Ethics Statement

For privacy and ethical reason, we deleted all personal information in example texts, including Figure 1, 3, and 4. Therefore, they differ from the original text. Judgments in two datasets used in this research are available to the members of the public once they have logged into China Judgments Online (https://wenshu.court.gov.cn/) and New Zealand Legal Information Institute (http://www.nzlii.org/). Nevertheless, to further protect the privacy of participants in the proceedings, when we publish our datasets, we will only publish the case reference number and URL instead of the full text. Furthermore, the published index datasets will only be allowed to be used for research reasons, not for any other purpose. To access the original judgments in China Judgment Online or New Zealand Legal Information Institute, please follow the websites' terms and conditions.

References

[1] Vidhisha Balachandran, Artidoro Pagnoni, Jay Yoon Lee, Dheeraj Rajagopal, Jaime Carbonell, and Yulia Tsvetkov, 'StructSum: Summarization via structured representations', in EACL, pp. 2575–2585, (2021).

[2] Katrina Banks-Smith, 'More than just precedent: Perspectives on judgment writing', volume 21, pp. 1–17. University of Notre Dame Fremantle, WA, (2019).

[3] Peter Butt, 'Judgment writing: an antipodean response', volume 129, pp. 7–10, (2013).

[4] Shuyang Cao and Lu Wang, 'Attention head masking for inference time content selection in abstractive summarization', in NAACL, pp. 5008–5016, (2021).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 'BERT: Pre-training of deep bidirectional transformers for language understanding', in NAACL, pp. 4171–4186, (2019).

[6] Mohamed Elaraby and Diane Litman, 'ArgLegalSumm: Improving abstractive summarization of legal documents with argument mining', in COLING, pp. 6187–6194, (2022).

[7] Filippo Galgani, Paul Compton, and Achim Hoffmann, 'Summarization based on bi-directional citation analysis', volume 51, pp. 1–24, (2015).

[8] Yan Gao, Zhengtao Liu, Juan Li, and Jin Tang, 'Extractive summarization of chinese judgment documents via sentence embedding and memory network', in NLPCC, eds., Lu Wang, Yansong Feng, Yu Hong, and Ruifang He, pp. 413–424, (2021).

[9] Alexios Gidiotis and Grigorios Tsoumakas, 'Structured summarization of academic publications', in Machine Learning and Knowledge Discovery in Databases, eds., Peggy Cellier and Kurt Driessens, pp. 636–645, (2020).

[10] José Ángel González, Annie Louis, and Jackie Chi Kit Cheung, 'Source-summary entity aggregation in abstractive summarization', in COLING, pp. 6019–6034, (2022).

[11] Nianlong Gu, Elliott Ash, and Richard Hahnloser, 'MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes', in ACL, pp. 6507–6522, (2022).

[12] Deepali Jain, Malaya Dutta Borah, and Anupam Biswas, 'Automatic summarization of legal bills: A comparative analysis of classical extractive approaches', in ICCCIS, pp. 394–400, (2021).

[13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer, 'BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension', in ACL, pp. 7871–7880, (2020).

[14] Shuaiqi Liu, Jiannong Cao, Ruosong Yang, and Zhiyuan Wen, 'Highlight-transformer: Leveraging key phrase aware attention to improve abstractive multi-document summarization', in ACL Findings, pp. 5021–5027, (2021).

[15] Shuaiqi Liu, Jiannong Cao, Ruosong Yang, and Zhiyuan Wen, 'Generating a structured summary of numerous academic papers: Dataset and method', in IJCAI, ed., Luc De Raedt, pp. 4259–4265, (2022).

[16] Yang Liu and Mirella Lapata, 'Hierarchical transformers for multi-document summarization', in ACL, pp. 5070–5081, (2019).

[17] Yang Liu and Mirella Lapata, 'Text summarization with pretrained encoders', in EMNLP, pp. 3730–3740, (2019).

[18] Yang Liu, Chenguang Zhu, and Michael Zeng, 'End-to-end segmentation-based news summarization', in ACL Findings, eds., Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, pp. 544–554, (2022).

[19] Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed Awadallah, and Dragomir Radev, 'DYLE: Dynamic latent extraction for abstractive long-input summarization', in ACL, pp. 1687–1698, (2022).

[20] Gianluca Moro and Luca Ragazzi, 'Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes', volume 36, pp. 11085–11093, (2022).

[21] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou, 'Summarunner: A recurrent neural network based sequence model for extractive summarization of documents', in AAAI, eds., Satinder Singh and Shaul Markovitch, pp. 3075–3081, (2017).

[22] Alec Radford and Karthik Narasimhan, 'Improving language understanding by generative pre-training', in OpenAI, (2018).

[23] Mathieu Ravaut, Shafiq Joty, and Nancy Chen, 'SummaReranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization', in ACL, pp. 4504–4524, (2022).

[24] Qian Ruan, Malte Ostendorff, and Georg Rehm, 'HiStruct+: Improving extractive text summarization with hierarchical structure information', in ACL Findings, pp. 1292–1308, (2022).

[25] Sajad Sotudeh and Nazli Goharian, 'TSTR: Too short to represent, summarize with details! intro-guided extended summary generation', in NAACL, pp. 325–335, (2022).

[26] Qiqi Wang, Kaiqi Zhao, Robert Amor, Benjamin Liu, and Ruofan Wang, 'D2GCLF: Document-to-graph classifier for legal document classification', in NAACL Findings, pp. 2208–2221, (2022).

[27] Wen Xiao and Giuseppe Carenini, 'Extractive summarization of long documents by combining global and local context', in EMNLP-IJCNLP, pp. 3011–3021, (2019).

[28] Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei, 'Sequence level contrastive learning for text summarization', pp. 11556–11565, (2022).

[29] Xingxing Zhang, Furu Wei, and Ming Zhou, 'HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization', in ACL, pp. 5059–5069, (2019).