Contents lists available at ScienceDirect

Information Processing and Management

Low-resource court judgment summarization for common law systems

Shuaiqi Liu [a,b], Jiannong Cao [a], Yicong Li [a,c], Ruosong Yang [a], Zhiyuan Wen [a,*]

[a] *The Hong Kong Polytechnic University, Kowloon, Hong Kong Special Administrative Region of China*
[b] *Huawei Technologies Co., Ltd, Shenzhen, China*
[c] *University of Technology Sydney, Sydney, Australia*

ARTICLE INFO

ABSTRACT

Common law courts need to refer to similar precedents' judgments to inform their current decisions. Generating high-quality summaries of court judgment documents can facilitate legal practitioners to efficiently review previous cases and assist the general public in accessing how the courts operate and how the law is applied. Previous court judgment summarization research focuses on civil law or a particular jurisdiction's judgments. However, judges can refer to the judgments from all common law jurisdictions. Current summarization datasets are insufficient to satisfy the demands of summarizing precedents across multiple jurisdictions, especially when labeled data are scarce for many jurisdictions. To address the lack of datasets, we present CLSum, the first dataset for summarizing multi-jurisdictional common law court judgment documents. Besides, this is the first court judgment summarization work adopting large language models (LLMs) in data augmentation, summary generation, and evaluation. Specifically, we design an LLM-based data augmentation method incorporating legal knowledge. We also propose a legal knowledge enhanced evaluation metric based on LLM to assess the quality of generated judgment summaries. Our experimental results verify that the LLM-based summarization methods can perform well in the few-shot and zero-shot settings. Our LLM-based data augmentation method can mitigate the impact of low data resources. Furthermore, we carry out comprehensive comparative experiments to find essential model components and settings that are capable of enhancing summarization performance.

1. Introduction

Common law systems rely on case precedents (prior cases), which encompass not only judgments within a particular jurisdiction but also judgments from all jurisdictions throughout the common law world (HKSAR DOJ, 0000).[1] Judges in common law jurisdictions need to find similar precedents and refer to the rationale employed in previous judgments (Bhattacharya et al., 2019). Court judgment documents typically contain long text that comprehensively discusses each case and provides detailed explanations of judges' decisions. Reading previous cases' judgments is crucial for legal practitioners in common law jurisdictions. There exist massive reported cases, and the number of cases is still increasing (HKSAR DOJ, 0000). It is difficult for legal practitioners to read through abundant cases' judgment documents.
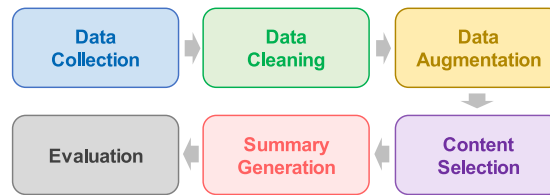

Fig. 1. Our workflow of court judgment summarization.

The high-quality summary of judgment document can facilitate readers to quickly browse key information of each case. It not only helps legal practitioners effectively review past cases, but also makes it easier for the general public to read judgments and understand how the courts operate and how the law is applied. However, employing legal experts to write summaries costs a lot and has limited coverage of new or atypical cases' judgments (Kanapala, Pal, & Pamula, 2019). Alternatively, we can leverage automatic text summarization technology to generate summaries for court judgments. Considering judges need to compare similar precedents across all common law jurisdictions (HKSAR DOJ, 0000), summaries of court judgments from multiple common law jurisdictions are helpful for comparing and analyzing similar precedents efficiently.

Our research focuses on empowering computers to produce high-quality summaries of multi-jurisdictional judgments even when lacking data and computational resources. To accomplish this goal, we must address several challenging issues: (1) the lack of datasets, (2) training supervised summarization models with very limited labeled data, (3) efficiently process long documents and summaries with limited computing resources, (4) accurately and comprehensively assess the quality of generated summaries.

We propose a solution for low-resource court judgment summarization to address the above challenging issues. Fig. 1 depicts six key steps in our solution, including data collection, data cleaning, data augmentation, content selection, summary generation, and evaluation.

There are very few court judgment summarization datasets. Some focus on civil law jurisdictions (Feijó & Moreira, 2018; Glaser, Moser, & Matthes, 2021), while others concentrate on judgments from specific common law jurisdictions (Bajaj et al., 2021; Shukla et al., 2022). They are insufficient to satisfy the demands of summarizing court judgment documents across multiple common law jurisdictions and comprehensively evaluating judgment summarization methods. In order to address the lack of datasets, we build CLSum, the first large-scale dataset for summarizing multi-jurisdictional common law court judgment documents.[2] CLSum has four subsets for court judgments from Canada, Australia, the United Kingdom, and Hong Kong SAR.

Similar to other domain-specific tasks, court judgment summarization usually suffers from a shortage of labeled data. Most judgments do not have summaries due to the high cost of hiring experts to write summaries. In most jurisdictions, court websites do not publish or only publish a small number of judgment summaries of typical cases. Under this low data resource condition, summarization models' few-shot and zero-shot performance becomes crucial. We carry out extensive comparative experiments to analyze the effect of the training set size on different summarization models' performance. In addition to selecting models with good few-shot and zero-shot performance, expanding the dataset is another way to improve summarization performance. To expand our CLSum's training sets and reduce overfitting, we adopt the large language model (LLM) for data augmentation. Meanwhile, we introduce legal knowledge into the prompts to constrain the LLM to properly use legal concepts when synthesizing sample text in the data augmentation process.

In addition to insufficient data resources, deploying artificial intelligence models in domain applications is often hindered by limited computing resources. Processing long documents and summaries usually requires more computing resources. When computing resources are limited, it is crucial to reduce the complexity of the summarization models and improve the efficiency of model training and inference. To reduce the complexity of the summarization models, our solution has a two-stage summarization framework, which first compresses long inputs in the content selection stage and then feeds the compressed inputs into the abstractive summarization models for fine-grained content selection and integration. Besides, we also substitute the original self-attention mechanism with sparse attention mechanisms (Beltagy, Peters, & Cohan, 2020; Dao, Fu, Ermon, Rudra, & Ré, 2022) and adopt a divide-and-conquer-based training strategy for models pre-trained on the shorter input sequences. To further reduce the consumption of GPU memory and conduct model fine-tuning and inference on off-the-shelf GPUs, we adopt some memory-efficient training techniques (Dettmers, Lewis, Belkada and Zettlemoyer, 2022; Dettmers, Lewis, Shleifer and Zettlemoyer, 2022; Dettmers, Pagnoni, Holtzman, & Zettlemoyer, 2023; Hu et al., 2021; Rajbhandari, Rasley, Ruwase, & He, 2020; Wu, Li, Aminabadi, Yao, & He, 2023).

It is also crucial to accurately and comprehensively evaluate the quality of the generated summaries. We adopt various summarization methods as baselines and evaluate them on our CLSum dataset. For performance comparison, we carry out automatic evaluation and human evaluation. Apart from the widely used evaluation metrics, we design a legal knowledge enhanced evaluation metric named LTScore. It is based on foundation models fine-tuned on the legal corpus. Legal texts usually contain more legal terms compared to texts in the general domain. These terms should be used accurately in court judgments and their summaries. Therefore, LTScore assigns greater weights to legal terms in judgment summaries to better assess the accurate usage of such legal terms.

---

[2] The CLSum dataset is available for download online at: github.com/StevenLau6/CLSum.

The rest of this paper is organized as follows. We list our objectives and contribution in Section 2. Section 3 briefly introduces related works. We present our CLSum dataset in Section 4, our summarization method in Section 5, and our experimental settings in Section 6. Our experimental results are reported and analyzed in Section 7. Finally, Section 8 concludes this paper and discusses our future work.

2. Objectives and contribution

The primary focus of this research is to generate high-quality summaries of judgment documents in the context of insufficient data and computational resources. To achieve this goal, we need to complete four objectives:

- To build a summarization dataset covering court judgment documents from multiple jurisdictions.
- To alleviate the impact of insufficient labeled data and improve the summarization model's performance.
- To effectively utilize limited computing resources for processing long judgment documents.
- To accurately and comprehensively evaluate the quality of generated court judgment summaries.

The contribution of this work is threefold:

- We present CLSum, the first large-scale dataset for summarizing common law court judgment documents from multiple jurisdictions.
- We are the first to employ large language models for data augmentation, summary generation, and evaluation in court judgment summarization.
- We design a legal knowledge enhanced evaluation metric named LTScore to evaluate generated legal text.

3. Related work

Automatic text summarization techniques aim to produce concise summaries that retain the salient information within input documents (Liu et al., 2023; Liu, Cao, Yang & Wen, 2022c). Both the publicly available datasets and advanced summarization methods facilitate the summarization research. Summarizing short documents (e.g., news and product reviews) has been widely studied (Fabbri, Li, She, Li, & Radev, 2019; Grusky, Naaman, & Artzi, 2018; Liu, Cao, Yang, & Wen, 2021; Zhuang, Jing, & Zhu, 2006). There has been increasing attention towards long document summarization in recent years. Researchers built some summarization datasets for long documents collected from various domains, including academic literature (Liu, 2024; Liu, Cao, Yang, & Wen, 2022a), government reports (Huang, Cao, Parulian, Ji, & Wang, 2021), and financial reports (Liu et al., 2023; Liu, Cao, Yang & Wen, 2022c).

3.1. Court judgment summarization datasets

There are very few court judgment summarization datasets. Some datasets focus on civil law jurisdictions (e.g., Brazil and Germany) (Feijó & Moreira, 2018; Glaser et al., 2021). While datasets for common law court judgments concentrate on specific jurisdictions. The Amicus Briefs dataset (Bajaj et al., 2021) focuses on public health cases in the United States.[3] Shukla et al. (2022) propose datasets summarizing court judgment documents from the United Kingdom and India.[4]

In a common law system, judges are required not only to consider similar local precedents but also to compare similar precedents in other jurisdictions. Summaries of court judgments from multiple jurisdictions can help judges compare and analyze similar precedents efficiently. However, the formats and content of court judgment documents vary across different jurisdictions. A summarization model that performs well on judgment documents from one jurisdiction may not exhibit effectiveness when applied to judgments from other jurisdictions. Current summarization datasets are insufficient to satisfy the demands of summarizing court judgment documents across multiple common law jurisdictions and comprehensively evaluating judgment summarization methods.

3.2. Court judgment summarization methods

Document summarization methods can be generally categorized into two types: extractive (Erkan & Radev, 2004; Li & Xu, 2023; Mihalcea & Tarau, 2004) and abstractive (Liu, Cao, Yang, & Wen, 2022b; Zhang, Zhao, Saleh, & Liu, 2020) summarization methods. We summarize and compare previous court judgment summarization methods in Table 1.

Extractive summarization methods are widely utilized in legal document summarization. They select the most salient input sentences to comprise summaries. LetSum (Farzindar & Lapalme, 2004) aims to produce structured summaries comprising four pre-defined themes. It uses heuristic rules to score and rank relevant sentences, and then select the sentences with the highest scores for each theme to comprise the structured summary. Similarly, Saravanan et al. (2006) employ the Conditional Random Field (CRF) to divide a legal document into seven labeled components. Then, they use the k-mixture model to select sentences for each component in the structured summary. CaseSummarizer (Polsley et al., 2016) ranks input documents' sentences based

---

[3] publichealthlawcenter.org/amicus-briefs.

[4] India's legal system comprises a blend of common law, civil law, equitable law, as well as customary and religious laws (Ashish & Mohit, 0000).

Table 1

Court judgment summarization methods. "Ext" and "Abs" stand for the extractive and abstractive summarization methods.

	Type	Models	Jurisdictions	Legal system
LetSum (Farzindar & Lapalme, 2004)	Ext	Heuristic	CA	Common law
CaseSummarizer (Polsley, Jhunjhunwala, & Huang, 2016)	Ext	Heuristic	Aus	Common law
Saravanan, Ravindran, and Raman (2006)	Ext	CRF	India	Mix
Feijó and Moreira (2018)	Ext	Gensim, LexRank, TextRank	Brazil	Civil law
Glaser et al. (2021)	Ext	CNN, RNN	Germany	Civil law
LegalSumm (Feijo & Moreira, 2023)	Abs	Transformer	Brazil	Civil law
Bajaj et al. (2021)	Abs	BART	US	Common law
Shukla et al. (2022)	Abs	BART, Pegasus, LED	UK, India	Mix
Ours	Abs	LLMs	HK, CA, UK, Aus	Common law

on their tf-idf scores (Salton & Buckley, 1988) coupled with some customized features. Feijó and Moreira (2018) compare some commonly used unsupervised extractive summarization methods (Gensim (Rehurek, 2010), LexRank (Erkan & Radev, 2004), and TextRank (Mihalcea & Tarau, 2004)). Glaser et al. (2021) employ convolutional neural networks (CNN) and recurrent neural network (RNN) based extractive summarization methods. Despite the notable development of extractive summarization methods over the last few decades, the extracted summaries still face difficulties in terms of coherence and readability (Wang, Nishino, Hirao, Sudoh, & Nagata, 2016; Yao, Wan, & Xiao, 2017). Consequently, abstractive summarization methods have received more attention in recent years.

Abstractive summarization methods identify the salient content in input text and generate novel sentences as summaries (Liu et al., 2022b). Unlike extractive summarization methods, there are relatively few abstractive summarization methods for court judgment documents. LegalSumm (Feijo & Moreira, 2023) divides the legal document into chunks and employs a transformer model (Vaswani et al., 2017) to generate these chunks' summaries. Then, it uses a BERT model (Devlin, Chang, Lee, & Toutanova, 2019) to assess these chunk summaries' faithfulness and select the one with the highest score as the final summary. Its output summaries are much shorter than the target summaries in our CLSum dataset. Shukla et al. (2022) employ some pre-trained sequence-to-sequence models to summarize court judgments. Bajaj et al. (2021) fine-tune the BART model (Lewis et al., 2020) on public health judgments in the United States.

There is a lack of research on comprehensively assessing summarization models' performance on multi-jurisdictional judgments. Most jurisdictions lack labeled data, which makes it difficult to train supervised summarization models. Previous legal document summarization work usually neglects abstractive summarization models' few-shot and zero-shot performance. In addition to the scarcity of labeled data resources, deploying artificial intelligence models in domain applications is often constrained by limited computing resources. There is a research gap in court judgment summarization under the constraints of low data and computing resources. Additionally, more research is needed to utilize legal knowledge in the abstractive summarization of court judgments.

4. CLSum dataset

Common law court judgment summarization (CLSum) is a large-scale summarization dataset covering court judgments from four common law jurisdictions: Canada, Australia, the United Kingdom, and Hong Kong SAR. This section first presents our procedures for collecting and pre-processing data. Subsequently, we describe four subsets in CLSum. Furthermore, we carry out statistics on CLSum and perform a comparative analysis with other datasets.

4.1. Collecting and pre-processing data

Court judgment documents usually comprehensively discuss each case and explain judges' decisions. Electronic files of judgments are usually publicly available online. In most jurisdictions, court websites do not publish or only publish a small number of judgment summaries of typical cases. We collect four jurisdictions' court judgment documents together with their summaries from court websites.

After collecting thousands of court judgments' HTML or PDF files, we parse these files and extract their content. Then, we conduct a series of data pre-processing operations, including eliminating noises, eliminating replicated examples and outliers with excessively short content, and splitting three sets for training, validation, and test.

4.2. Description of the CLSum's subsets

We collect multi-jurisdictional common law court judgment documents to build the CLSum dataset. CLSum consists of four subsets: CLSum-CA, CLSum-HK, CLSum-UK, and CLSum-AUS.

CLSum-CA is collected from the website of the Supreme Court of Canada (SCC).[5] We collect the case briefs and corresponding judgment documents from 2018 to 2023. CLSum-CA has the fewest samples among these subsets.

---

[5] www.scc-csc.ca/case-dossier/cb/index-eng.aspx.

Table 2

Summarization datasets' statistical information. "Samples" is the sample number in the dataset. "Doc" and "Sum" stand for the input document and target summary. "Sents" and "Words" represent the average number of sentences and words in each input document or target summary. "Dens." and "Cov." are the density and coverage of extractive fragments.

Dataset	Samples	Sents (Doc)	Words (Doc)	Sents (Sum)	Words (Sum)	Dens.	Cov.
CNN/DM	312,085	39.8	810.6	3.7	56.2	3.8	0.9
PubMed	133,215	87.5	3049.0	6.8	202.4	5.8	0.8
arXiv	215,913	205.7	6029.9	9.6	272.7	3.8	0.9
CLSum-CA	192	1168	38,403	38	748	0.8	0.5
CLSum-HK	233	395	11,911	46	1169	9.7	0.9
CLSum-UK	793	458	16,143	41	1241	2.4	0.7
CLSum-AUS	1019	630	20,485	19	592	1.4	0.6

Table 3

The percentage of target summaries' new n-grams.

Dataset	Unigrams	Bigrams	Trigrams	4-grams
arXiv	15.04	48.21	71.66	83.26
PubMed	18.38	49.97	69.21	78.42
CNN/DM	19.50	56.88	74.41	82.83
CLSum-CA	**21.65**	**58.00**	**80.90**	**90.09**
CLSum-HK	13.48	38.86	57.53	69.06
CLSum-UK	15.00	36.25	53.71	64.29
CLSum-AUS	11.65	37.69	58.01	70.53

CLSum-HK is collected from the legal reference system.[6] It covers typical cases from multilevel courts, including the Coroner's Court, Magistrates' Court, District Court (DC), High Court (HC), and Court of Final Appeal (CFA) in Hong Kong. We collect these cases' judgment documents and their press summaries from 2012 to 2023.

CLSum-UK is the subset focusing on the United Kingdom Supreme Court's judgment documents.[7] Shukla et al. (2022) collected British judgment documents and their press summaries from 2009 to 2021.

CLSum-AUS is collected from the High Court of Australia's website.[8] We collect Australian judgment documents and their summaries from 2005 to 2023. CLSum-AUS is the subset with the most samples.

4.3. Dataset analysis

We conduct statistics on CLSum's four subsets and perform a comparative analysis with other datasets. As shown in Table 2, these four subsets' input documents and target summaries are much longer in comparison to existing datasets. The formats and content of court judgments vary across these four subsets. CLSum-HK and CLSum-CA have a few samples, while CLSum-UK and CLSum-AUS have more samples. Among these four subsets, CLSum-CA and CLSum-AUS have longer input documents but shorter target summaries.

In order to quantify the abstraction level of CLSum's target summaries, Table 3 counts the ratio of target summaries' n-grams that are not present in the inputs. Target summaries of CLSum-CA exhibit a greater number of new n-grams and a higher abstraction level. The abstraction level of target summaries in CLSum-HK, CLSum-UK, and CLSum-AUS is comparatively lower than that in other datasets.

Additionally, we utilize two measures (Grusky et al., 2018), including the coverage and density of extractive fragments, to assess summarization datasets' extractive nature. The set of shared token sequences in a given document D and its summary S is denoted as extractive fragments $F(D, S)$. Eq. (1a) calculates the extractive fragment coverage, representing the proportion of words in the summary that are part of the extracted segment in the input document. Eq. (1b) computes the extractive fragment density, which measures the average length of the extraction fragment to which each word in the summary belongs. Besides, the ratio between the number of words in the document and its summary is the compression ratio (Grusky et al., 2018).

$$\text{COVERAGE}(D, S) = \frac{1}{|S|} \sum_{f \in F(D,S)} |f| \tag{1a}$$

$$\text{DENSITY}(D, S) = \frac{1}{|S|} \sum_{f \in F(D,S)} |f|^2 \tag{1b}$$

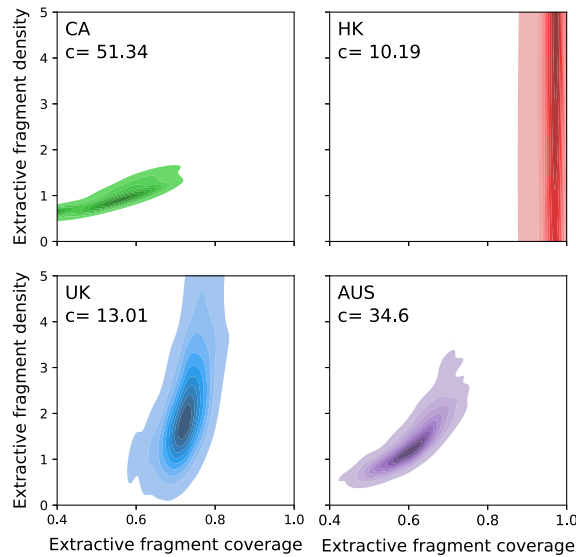$$\text{COMPRESSION}(D, S) = \frac{|D|}{|S|} \tag{1c}$$

Fig. 2. Distributions of extractive fragment coverage and extractive fragment density. "c" denotes the compression ratio.

As shown in Table 2, CLSum-HK's coverage is similar to previous summarization datasets but is higher than that of other subsets in CLSum. Among these four subsets, CLSum-CA and CLSum-AUS have smaller coverage and density of extractive fragments. Fig. 2 depicts the visualization of distributions of two measures using kernel density estimation. CLSum-HK and CLSum-UK subsets have high variability in density, which suggests their target summaries are written in varying styles. Furthermore, the compression ratio is calculated by dividing the word count of a document by that of its corresponding summary.

5. Method

Fig. 1 depicts that our solution consists of six key steps: data collection, data cleaning, data augmentation, content selection, summary generation, and evaluation. Our data collection and cleaning procedures are introduced in Section 4.1. After the first two steps, we conduct data augmentation to expand the training sets and reduce overfitting to the limited training samples. Then, the content selection and summary generation steps complete the selection and integration of key information from rough to fine. Our evaluation step comprises both automatic evaluation and human evaluation for assessing the summaries generated by various summarization models.

To the best of our knowledge, this is the first court judgment summarization work adopting LLM in data augmentation, summary generation, and evaluation. Summarizing court judgment documents under low data and computing resources conditions has several challenging issues, including: training supervised models with extremely limited labeled data, identifying the salient content dispersed within the long judgment document, and improving the efficiency of model training and inference to process long input documents and summaries. This section presents our methods to address these aforementioned issues.

5.1. Mitigating the impact of insufficient labeled samples

In most jurisdictions, courts do not publish or only publish a limited number of judgment summaries for typical cases. The limited size of the labeled training set usually hinders the performance of supervised models trained from scratch. It is essential to guarantee the summarization model's performance when generalizing to cases not seen during training. Labeling large-scale datasets can cost a lot, while unlabeled data can be easily collected from the Internet. Researchers pre-trained foundation models with self-supervised tasks on massive unlabeled data to learn better text representations. These foundation models can provide good initialization and reduce the amount of labeled training samples required for downstream tasks. By fine-tuning on the downstream task, these foundation models often outperform models trained on the same task from scratch. To mitigate the impact of insufficient labeled samples from the summarization model perspective, we evaluate diverse foundation models' few-shot and zero-shot performance on our CLSum dataset and select the best-performing model.

Meanwhile, we also study mitigating the impact of insufficient labeled samples from the data perspective. We propose knowledge-constrained rephrasing, an LLM-based data augmentation method constrained by legal knowledge. We also compare it with different data augmentation methods like back translation and rephrasing. These data augmentation methods can expand the training sets and reduce overfitting to the limited training samples. The back translation is a commonly used data augmentation method. It first translates the text into another language (e.g., from English to German) and then translates it back to the original language (e.g., from German to English) (Sennrich, Haddow, & Birch, 2016). The rephrasing method employs large language models to
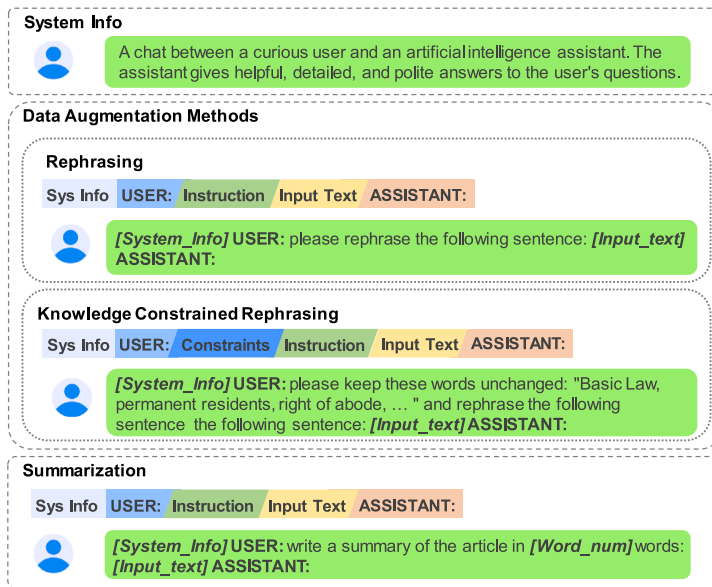
Fig. 3. Prompt templates and examples for large language models.

Table 4
Evaluation results of content selection methods. "R-1" is the unigram recall, and "R-avg" represents the mean value of the recalls of unigram, bigram, trigram, and 5-gram. "Lead" represents the truncation method.

Method	CLSum-CA		CLSum-HK		CLSum-UK		CLSum-AUS	
	R-1	R-avg	R-1	R-avg	R-1	R-avg	R-1	R-avg
Lead	68.41	30.46	85.67	52.29	**83.81**	**54.47**	80.05	46.95
LexRank	69.10	30.61	85.61	52.38	83.15	53.84	85.83	50.91
TextRank	**69.88**	**30.88**	**85.68**	**52.45**	83.06	53.70	**85.90**	**50.91**

rephrase each sentence in judgment documents and target summaries (Dai et al., 2023). Legal texts usually contain more legal terms compared to texts in the general domain. These terms must be used accurately in court judgments and their summaries. Therefore, we propose knowledge-constrained rephrasing, which introduces legal knowledge into the prompts of LLMs to constrain the synthesized sentences to accurately use legal concepts in the data augmentation process. The prompt templates and examples of these data augmentation methods are shown in Fig. 3. These generated sentences are merged as new data samples. We supplement the synthetic data into the training sets to alleviate the impact of insufficient labeled data.

5.2. Salient content identification and integration

Judgment documents in our CLSum datasets usually contain tens of thousands of words, as depicted in Table 2. The salient content is dispersed within different parts of these long judgment documents. Nonetheless, foundation models are commonly pre-trained on text sequences that have a predetermined maximum length. When abstractive summarization models cannot accept the entire document as input, compressing the input length while preserving key information is crucial. Apart from simply truncating the document, there exist more efficient content selection methods.

We conduct two-stage operations to identify and integrate the salient content from coarse to fine. The first stage, named salient content selection, can be regarded as a rough selection. The recall of essential content that should be retained in summaries is maximized during the compression of long inputs. Subsequently, the condensed inputs are passed to summarization models for fine-grained content selection and integration. The salient content selection stage and summarization stage are shown in the upper part of Fig. 4.

The step of content selection is designed to preserve the maximum key information when compressing the input to a fixed length. We compare different methods' performance and mainly focus on their average recall of n-grams. LexRank and TextRank methods score and rank input sentences and then select the top-ranked sentences according to a predefined length.[9] Table 4 shows these methods' evaluation results on the training set of CLSum. We choose the most effective method for content selection in our subsequent experiments. Specifically, we adopt TextRank for CLSum-CA, CLSum-HK, and CLSum-AUS and use truncation for CLSum-UK.

---

[9] In our experiments, content selection methods compress the input length to 16,384 tokens.
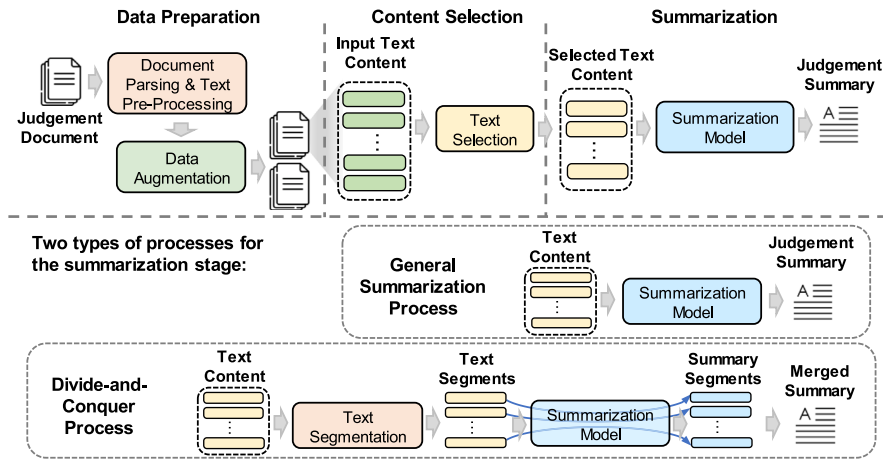
Fig. 4. Detailed workflow of our method. Two detailed processes of the summarization stage in the upper part of the figure is shown in the lower part of this figure.

5.3. Improving the efficiency of models and training methods

Most real-world applications not only face low data resources but also have the constraint of low computing resources. Especially when available computing resources are limited, improving the efficiency of model training and inference is a crucial issue. Many summarization methods require large computing resources when processing long documents, which limits their applications. In transformer-based models (Vaswani et al., 2017), the self-attention mechanism's complexity exhibits a quadratic increase with the input length. It can take up a lot of GPU memory and limit models' efficiency. Moreover, the limited GPU memory size poses constraints on transformer-based models' capability to model longer context. To improve summarization models' efficiency, we employ sparse attention mechanisms (Beltagy et al., 2020; Dao et al., 2022; Guo et al., 2022). Summarization models employing sparse attention mechanisms have the capability to model longer contexts with the same size of GPU memory. Additionally, our two-stage summarization framework also reduces the context length that neural summarization models need to model, thus reducing the associated GPU memory consumption.

In addition to efficient models, efficient training methods can also expedite the training process. We adopt some memory-efficient training methods, like gradient accumulation, gradient checkpointing,[10] parameter quantization (Dettmers, Lewis, Belkada et al., 2022; Wu et al., 2023), memory-efficient optimizer (Dettmers, Lewis, Shleifer et al., 2022; Rajbhandari et al., 2020), and adding parameter-efficient adapters (Dettmers et al., 2023; Hu et al., 2021). For those models pre-trained on the shorter input sequences, we adopt a divide-and-conquer-based training strategy for generating summary segments, followed by merging them to form the final summary. The general process for models capable of modeling long texts and the divide-and-conquer-based process for models pre-trained on shorter texts are shown in the lower part of Fig. 4. These efficient training methods enable us to fine-tune LLMs on lengthy inputs using one off-the-shelf GPU.

6. Experiments

6.1. Baselines

We employ various summarization methods as baselines and evaluate them on our CLSum dataset.

TextRank and LexRank (Erkan & Radev, 2004; Mihalcea & Tarau, 2004) are graph-based ranking methods that are widely employed in unsupervised extractive summarization.

Longformer-Encoder-Decoder (LED) (Beltagy et al., 2020) is built on the architecture of the BART model (Lewis et al., 2020) and employs sparse attention mechanisms to replace the original self-attention mechanism within its encoder.

Legal-LED[11] is the LED model fine-tuned on the litigation releases of U.S. Securities and Exchange Commission (SEC).[12]

LongT5 (Guo et al., 2022) replaces the self-attention mechanism with a global–local attention mechanism in the encoder part of T5 model (Raffel et al., 2020) to model longer inputs.

BLOOM (Workshop et al., 2022) is an open-source LLM collection with parameter numbers ranging from 560 m to 176B. BLOOM is pre-trained on a corpus comprised of dozens of languages.

LLaMA (Touvron et al., 2023) is a collection of LLMs whose parameter numbers range from 7B to 65B. LLaMA is trained on publicly available data.

Vicuna (Chiang et al., 2023) is a set of LLaMA models fine-tuned with user-shared ChatGPT conversation data.

GPT-3.5-turbo[13] is the model employed in the ChatGPT. Its fine-tuning process employs Reinforcement Learning from Human Feedback (RLHF) on the GPT-3.5 model (OpenAI, 2022).

6.2. Experimental setting

For LED-based models (LED and Legal-LED), the vocabulary size is set as 50,265, whereas LLaMA-based models (Vicuna and LLaMA) and LongT5 model utilize a default vocabulary size of 32,000 and 32,128, respectively. When fine-tuning the LED-based model, we set the learning rate to $5e^{-5}$. The LLaMA-based models and LongT5 model use $2e^{-5}$ and $1e^{-3}$, respectively. We utilize the warmup and decay of the learning rate for all these models. As for the optimizer, we use Adam (Kingma & Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for LED-based models and Adafactor (Shazeer & Stern, 2018) for T5-based models. When fine-tuning LLaMA and Vicuna models, we use 4-bit NormalFloat (NF4), QLoRA, and 32-bit paged AdamW optimizer (Dettmers et al., 2023) to save GPU memory. Different foundation models are pre-trained on texts of different lengths. In the fine-tuning stage, we predefine the maximum input length for each model to match its input length during the pre-training. Given the constraints of GPU memory size, models equipped with sparse attention mechanisms (e.g., LongT5, LED, Legal-LED) can be pre-trained on longer text sequences. Their maximum input length is 16,384. We fine-tuned them to generate the full summary directly. For models pre-trained on the shorter input sequences (e.g., LLaMA and Vicuna), we utilize the divide-and-conquer-based training strategy for generating summary segments, followed by combining them to form the final summary. We employ the beam search, whose beam size is five. We utilize the implementations of LongT5, LED, Legal-LED, and Vicuna from HuggingFace's Transformers (Wolf, Chaumond, Debut, et al., 2020) and LLaMA's implementation from Touvron et al. (2023). We fine-tune these models using one GPU named Nvidia RTX8000.

6.3. Evaluation metrics

We carry out automatic evaluation and human evaluation for assessing the summaries generated by various models. The automatic evaluation metrics we used can be further divided into statistics-based evaluation metrics (e.g., ROUGE) and model-based evaluation metrics (e.g., BARTScore). We not only employ these commonly used evaluation metrics but also propose a novel evaluation metric named legal text score (LTScore).

We present F_1 scores of Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004) in our experimental results. Specifically, we measure overlaps of unigrams (R-1), bigrams (R-2), and the longest common subsequence (R-L) between output summaries and target summaries.

BARTScore (Yuan, Neubig, & Liu, 2021) is a model-based evaluation metric assessing the quality of generated text by formulating it as a text generation task. Built on the pre-trained BART model (Lewis et al., 2020), BARTScore calculates the log probability of one text sequence y when given another text sequence x. In Eq. (2), θ represents the given pre-trained BART model's parameters. BARTScore sets equal weight ω_t for each token.

$$\text{BARTScore} = \sum_{t=1}^{m} \omega_t \log p\left(\mathbf{y}_t \mid \mathbf{y}_{<t}, \mathbf{x}, \theta\right) \tag{2}$$

Compared with general documents, legal documents usually contain many specialized expressions and domain knowledge. Compared with the BART model trained on the general domain corpus, the models trained on legal instruments have a better command of these specialized expressions and domain knowledge. To evaluate the generated legal text, we design an evaluation metric named legal text score (LTScore). LTScore employs a set of foundation models (e.g., LED and Vicuna) fine-tuned on our legal corpus to vote the score for each text sequence.

Legal texts usually contain more legal terms compared to texts in the general domain. These terms must be used accurately in court judgments and their summaries. Therefore, LTScore assigns greater weight to legal terms in judgment summaries to better evaluate whether these legal terms are used accurately. LTScore can be calculated according to the following formulas.

$$\text{LTScore}_P = \sum_{j=1}^{n} \omega'_j \sum_{t=1}^{m} \omega_t \log p\left(\text{cand}_t \mid \text{cand}_{<t}, \text{ref}, \theta_j\right) \tag{3a}$$

$$\text{LTScore}_R = \sum_{j=1}^{n} \omega'_j \sum_{t=1}^{m} \omega_t \log p\left(\text{ref}_t \mid \text{ref}_{<t}, \text{cand}, \theta_j\right) \tag{3b}$$

$$\text{LTScore}_{F_1} = \frac{2 \times \text{LTScore}_P \times \text{LTScore}_R}{\text{LTScore}_P + \text{LTScore}_R} \tag{3c}$$

$$\omega_t = \begin{cases} 1, & \text{if token}_t \notin g_i \\ 1 + e^{\omega_{g_i}}, & \text{if token}_t \in g_i \end{cases} \tag{3d}$$

---

[13] We adopt the GPT-3.5-turbo-0301 API from Azure Cloud.

Table 5

Automatic evaluation results of summarization models' zero-shot performance. R1, R2, and RL are the F_1 score of ROUGE-1, ROUGE-2, and ROUGE-L.

Method	CLSum-CA R1/R2/RL	CLSum-HK R1/R2/RL	CLSum-UK R1/R2/RL	CLSum-AUS R1/R2/RL
LexRank	31.87/9.54/13.24	49.66/23.58/21.41	60.28/26.86/22.84	53.57/24.46/24.24
TextRank	32.03/9.36/13.57	51.50/24.36/23.65	**60.62**/27.22/25.39	54.31/24.51/24.61
GPT3.5	**50.01/18.58/20.62**	**54.28**/24.04/**25.13**	57.05/25.51/24.10	54.10/25.51/25.11
BLOOM$_{560M}$	38.83/11.52/16.41	42.31/14.92/19.52	49.29/18.15/20.71	37.80/14.74/17.84
BLOOM$_{7B1}$	39.05/11.90/16.78	46.07/18.10/20.97	53.75/22.36/23.15	40.29/17.43/19.06
LLaMA$_{7B}$	39.88/11.90/15.99	47.60/18.22/20.91	54.72/22.52/22.59	40.55/15.75/18.88
LLaMA$_{13B}$	40.59/12.63/16.19	48.21/18.90/20.88	52.80/21.87/22.23	43.41/17.44/19.74
Vicuna$_{7B}$	47.32/16.42/20.00	53.01/23.20/23.94	57.29/27.38/24.40	57.27/**27.53/27.03**
Vicuna$_{13B}$	47.69/17.17/20.29	53.08/**24.45**/24.91	57.86/**28.69**/25.46	**57.33**/27.36/26.75
LongT5	23.29/6.08/10.23	46.73/16.63/19.32	52.32/19.68/20.40	43.43/16.93/19.88
LED$_{Base}$	23.63/6.83/11.58	47.26/18.16/19.85	56.08/21.52/21.43	44.84/15.73/21.25
Legal-LED	37.10/6.97/16.66	39.43/9.79/17.88	37.46/10.07/17.05	42.28/11.08/19.78
LED$_{Large}$	23.87/6.80/10.79	47.03/17.27/19.97	49.02/17.91/20.85	43.64/15.57/20.40

$$\omega_{g_i} = \frac{\text{Score}(g_i) - \text{Score}(G)_{\min}}{\text{Score}(G)_{\max} - \text{Score}(G)_{\min}} \qquad g_i \in G \tag{3e}$$

Eqs. (3a) and (3b) calculate the precision and recall of LTScore. In Eq. (3c), the F_1 score of LTScore is the arithmetic average of its recall and precision. We adopt a set of foundation models (LED and Vicuna) fine-tuned on our CLSum dataset to vote the final score in Eq. (3a) and (3b). Each fine-tuned foundation model calculates $\log p$ independently. The final LTScore is the weighted sum of the scores calculated by these models. $\theta_j \in \{\theta_1, \ldots, \theta_n\}$ represents the jth fine-tuned foundation model's parameters. In our experiments, we assign the equal weights $\omega'_j = 1/n$ for these fine-tuned foundation models.

To emphasize the precise use of legal terms, we assign different weights ω_t for input tokens. For each sample, we select the phrases appearing in the candidate sequence or reference sequence from the glossary of legal terms.[14] We rank these selected phrases according to their importance scores and then select the set of phrases G with top-100 importance scores $\text{Score}(G)$. In our experiments, we employ these phrases' tf-idf scores as their importance scores $\text{Score}(G)$. The g_i is the ith phrase in the selected top-100 phrases set G. Eq. (3e) calculates the Min–Max normalized importance score of g_i as ω_{g_i}. Eq. (3d) calculates the weight ω_t of the t-th token token_t in the candidate sequence or reference sequence. If the t-th token token_t is a part of the phrase g_i, we add the exponential weight of phrase g_i to the t-th token's weight w_t.

LTScore enhances the adaptation to legal texts from two aspects: not only by injecting legal knowledge through fine-tuning base models on the legal dataset but also by adjusting token weights to emphasize the precise use of legal terms.

7. Results and discussion

In this section, we exhibit our experimental results, and then we analyze and discuss these results. We carry out automatic evaluation and human evaluation for assessing the summaries generated by various summarization models. Furthermore, we carry out comprehensive comparative experiments to find essential model components and settings that are capable of improving summarization performance.

7.1. Summarization results

We employ multiple metrics to assess the quality of the output summaries in the automatic evaluation. Specifically, we adopt the F_1 score of ROUGE (Lin, 2004),[15] and some model-based metrics, including BARTScore$_{F_1}$ and our proposed LTScore$_{F_1}$. We evaluate different summarization models' zero-shot and few-shot performance.

Table 5 reports the zero-shot performance of different summarization models. Under the zero-shot setting, LLMs (GPT-3.5-turbo and Vicuna) are competitive on all subsets of our CLSum dataset. The Vicuna models (Chiang et al., 2023) fine-tuned with user-shared ChatGPT conversations largely surpass the original LLaMA models (Touvron et al., 2023) and the BLOOM model with similar parameter sizes. The zero-shot performance of some pre-trained sequence-to-sequence models with hundreds of millions of parameters (LongT5, LED, Legal-LED, and BLOOM$_{560M}$) is not as good as that of unsupervised extractive methods (LexRank and TextRank).

As for the few-shot setting, we fine-tune summarization models on training sets of increasing size (from ten examples to hundreds of examples). Figs. 5, 6, and 7 visualize the impact of training set size on the ROUGE-2 scores, BARTScore, and LTScore of the generated summaries. Even fine-tuning on only a few examples can bring obvious performance gains for these abstractive summarization methods, which validates the necessity of fine-tuning on downstream tasks. These summarization models' performance improvements decay as the training set size increases.

---

[14] www.glossary.doj.gov.hk/.

[15] github.com/bheinzerling/pyrouge.

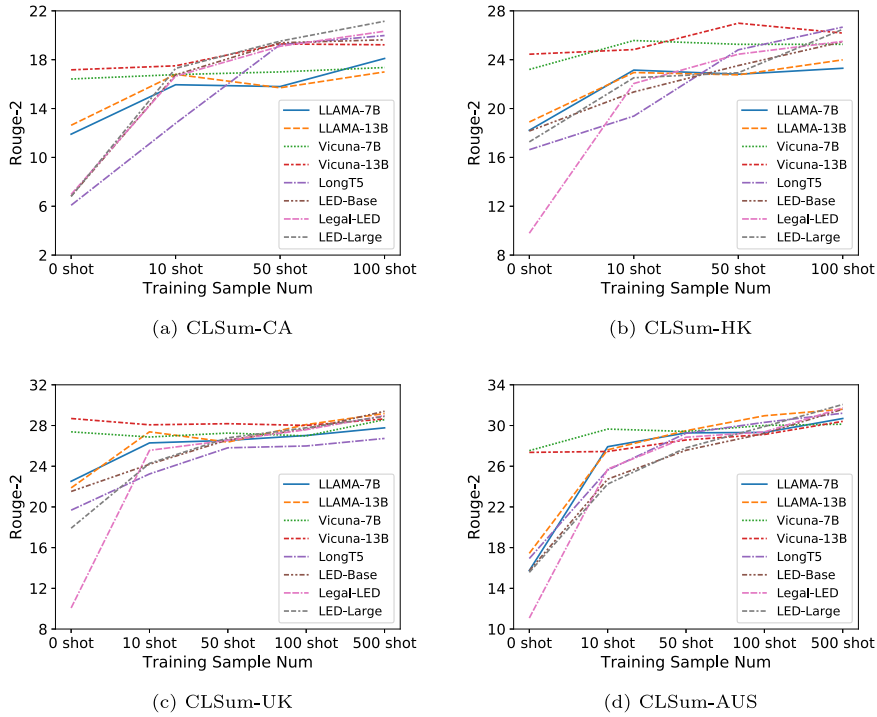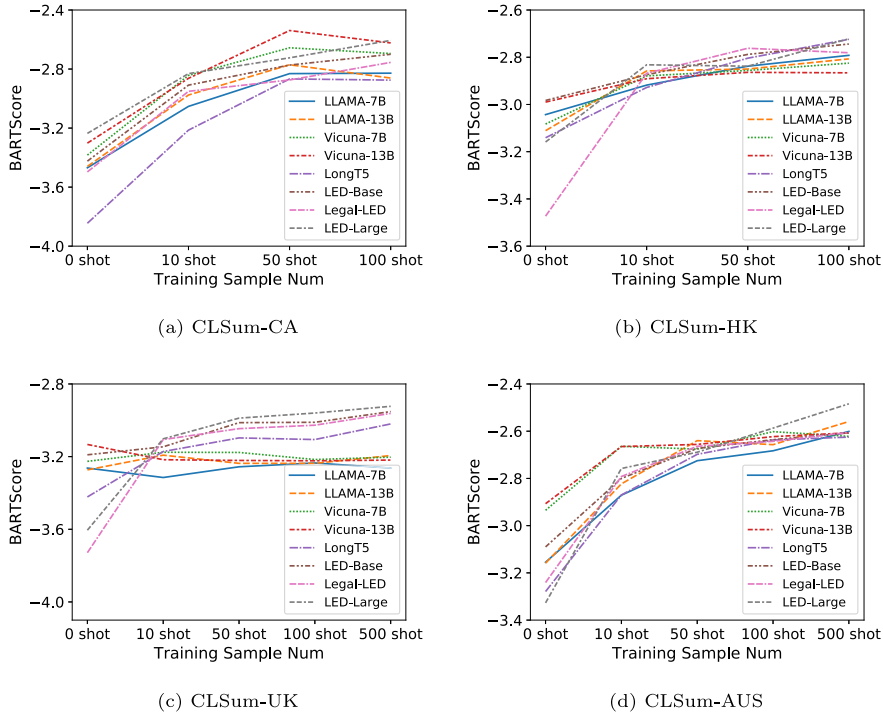**Fig. 5.** Automatic evaluation result (ROUGE-2_{F_1} Score) on CLSum.

Fig. 6. Automatic evaluation result (BARTScore$_{F_1}$) on CLSum.

In our human evaluation, we compare the outputs of summarization models based on their informativeness (i.e., cover salient content of input documents), fluency (i.e., summary content is well organized and uses grammar appropriately), and non-redundancy
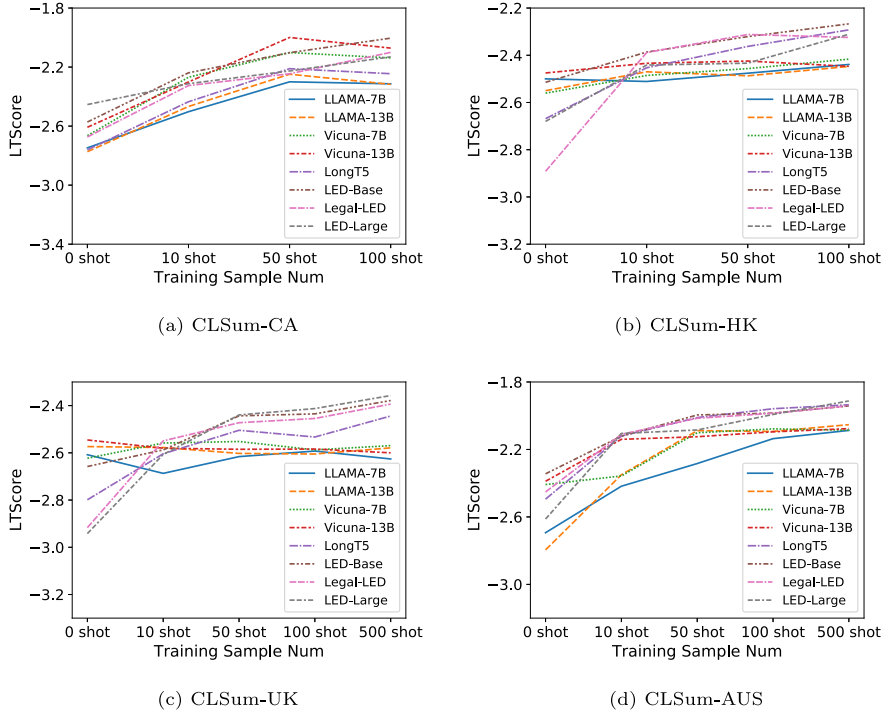
(a) CLSum-CA (b) CLSum-HK

(c) CLSum-UK (d) CLSum-AUS

**Fig. 7.** Automatic evaluation result ($LTScore_{F_1}$) on CLSum.

Table 6

Human evaluation results on CLSum dataset. "win" denotes that the current model's output summary surpasses that of LED_{Large} model in one dimension.

	$Vicuna_{13B}$				$Vicuna_{7B}$			
	win	lose	tie	kappa	win	lose	tie	kappa
CLSum-CA								
Informativeness	23.3%	31.1%	45.6%	0.671	21.1%	26.7%	52.2%	0.618
Fluency	18.9%	21.1%	60.0%	0.623	17.8%	18.9%	63.3%	0.603
Non-redundancy	27.8%	36.7%	35.6%	0.614	28.9%	33.3%	37.8%	0.648
CLSum-HK								
Informativeness	24.4%	28.9%	46.7%	0.635	26.7%	27.8%	45.6%	0.638
Fluency	20.0%	22.2%	57.8%	0.634	21.1%	24.4%	54.4%	0.629
Non-redundancy	17.8%	18.9%	63.3%	0.624	18.9%	20.0%	61.1%	0.677
CLSum-UK								
Informativeness	37.8%	30.0%	32.2%	0.648	35.6%	31.1%	33.3%	0.649
Fluency	28.9%	25.6%	45.6%	0.655	26.7%	24.4%	48.9%	0.612
Non-redundancy	17.7%	20.0%	62.2%	0.672	15.6%	16.7%	67.8%	0.636
CLSum-AUS								
Informativeness	30.0%	28.9%	41.1%	0.662	28.9%	27.8%	43.3%	0.625
Fluency	26.7%	24.4%	48.9%	0.647	23.3%	22.2%	54.4%	0.629
Non-redundancy	25.6%	26.7%	47.8%	0.668	20.0%	22.2%	57.8%	0.615

(i.e., less repetition in output summary). We selected 30 samples at random from each CLSum subset's test set. For each sample, three annotators assess and compare the anonymously presented output summaries from two models. Additionally, we evaluate the agreement among annotators using Fleiss' kappa (Fleiss, 1971).

Table 6 exhibits our human evaluation results. Three models fine-tuned on each entire subset are compared here. The CLSum-CA and CLSum-HK subsets have very few samples in their training sets. On these two subsets, the Vicuna models perform worse than the LED model in terms of informativeness. Tables 2 and 3 present that target summaries' average length is shorter in the CLSum-CA subset. The shorter target summaries in the CLSum-CA subset comprise more new n-grams that are absent in the input and exhibit lower coverage and density of extractive fragments. Generating these more abstractive summaries can be difficult, particularly when the summarization model is fine-tuned on a very small training set. We discover that Vicuna models trained with the divide-and-conquer method on the CLSum-CA subset generate more redundant and less informative summary content than the LED model. When there
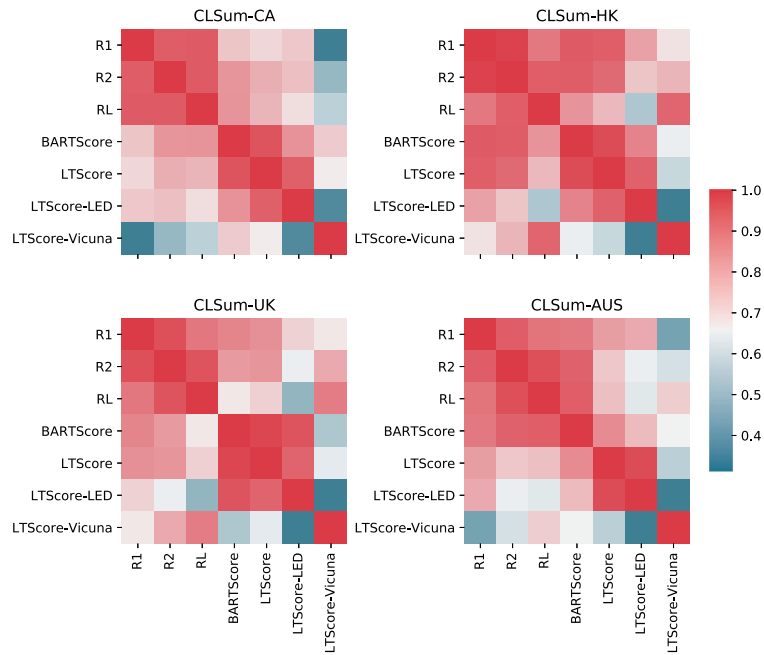
Fig. 8. Correlation of automatic evaluation metrics. "LTScore-LED" and "LTScore-Vicuna" are LTScore$_{F_1}$ calculated by a single finetuned foundation model. "LTScore" denotes the final weighted sum of LTScore$_{F_1}$. R1, R2, and RL are the F_1 score of ROUGE-1, ROUGE-2, and ROUGE-L. "BARTScore" represents BARTScore$_{F_1}$.

is a lack of training samples, the semantics of the generated summaries for different segments become relatively concentrated and exhibit more repeated content. Tables 2 and 3 also exhibit that the average length of target summaries is longer in the CLSum-HK subset. These longer target summaries' content is more diverse and less abstractive. When employing divide-and-conquer, semantically dispersed and longer target summaries for each segment will guide the summarization model to focus on different content when summarizing different segments. The lack of training samples in the CLSum-HK subset mainly affects informativeness and has less impact on the redundancy of generated summaries. On the CLSum-UK and CLSum-AUS subsets, the Vicuna models can outperform the LED model in informativeness, while these models are comparable regarding fluency and non-redundancy. These results verify that the training set size can affect the model acquiring the capacity to effectively summarize key information during fine-tuning, consequently influencing the informativeness of output summaries.

Fig. 8 depicts the Pearson correlation among various evaluation metrics. The correlation among ROUGE scores on different N-grams is high. Although the LTScores calculated by the individual fine-tuned foundation model (LED and Vicuna) have a lower correlation with ROUGE scores, the final LTScore (weighted sum) correlated with the ROUGE scores roughly as well as the BARTScore.

7.2. Discussion on the training set size

For models trained on the same training set, the larger models' few-shot performance is not always better than that of the smaller models. With the increase in the number of training samples, the performance of LLMs (LLaMA and Vicuna) improves slower than these smaller pre-trained sequence-to-sequence models (LongT5, LED, and Legal-LED). This may be caused by two reasons: (1) When adopting the QLoRA (Dettmers et al., 2023) technology, the number of trainable parameters is smaller than the entire model's parameter number. The small number of trainable parameters limits the new knowledge that the model can learn during fine-tuning; (2) Compared with the training data utilized in the pre-training stage and supervised fine-tuning (SFT) stage, the amount of labeled samples used in our fine-tuning process is relatively small, thereby having limited impact on model performance.

Finally, the performance of pre-trained models with hundreds of millions of parameters (LongT5, LED, and Legal-LED) can exceed that of a large language model with billions of parameters (LLaMA and Vicuna). Training smaller models with more labeled data can achieve comparable performance, which is critical to reducing the cost of deploying models in real-world applications.

7.3. Discussion on supervised fine-tuning and RLHF

The Vicuna models (Chiang et al., 2023) fine-tuned with user-shared ChatGPT conversations largely surpass the original LLaMA models (Touvron et al., 2023) in the zero-shot setting. Regarding the few-shot performance, supervised fine-tuning (SFT) can

Table 7
Evaluation results of summarization models trained on augmented datasets. R1, R2, and RL are the F_1 score of ROUGE-1, ROUGE-2, and ROUGE-L.

Method	Full train set R1/R2/RL	Rephrasing R1/R2/RL	Constrained rephrasing R1/R2/RL	Back translation R1/R2/RL
CLSum-CA				
LLaMA-7B	47.91/18.10/20.74	46.41/16.79/21.24	52.17/19.46/22.39	39.91/12.06/21.18
LLaMA-13B	48.09/17.00/20.45	45.90/17.09/21.56	51.02/19.41/22.70	47.21/17.30/21.94
Vicuna-7B	47.62/17.36/22.05	47.92/17.47/**23.10**	52.45/19.65/22.80	43.22/14.92/21.48
Vicuna-13B	50.66/19.22/22.68	49.93/18.86/23.06	51.02/18.49/21.79	49.39/18.70/22.36
LongT5	55.85/19.98/21.48	55.01/19.88/21.73	55.31/20.18/21.84	55.62/19.93/21.70
LED-Base	54.57/19.63/21.32	53.28/19.57/21.39	54.94/20.08/22.10	52.75/19.29/20.80
Legal-LED	56.04/20.33/21.73	53.95/19.95/21.63	54.95/20.64/22.13	55.09/20.29/21.56
LED-Large	**57.23**/21.15/22.65	56.64/21.17/22.17	56.62/**21.72**/22.81	56.50/21.00/22.28
Average improvement %		−2.14/−1.32/1.65	**2.82**/**4.82**/**3.29**	−4.62/−6.41/0.20
CLSum-HK				
LLaMA-7B	51.71/23.30/26.18	52.14/23.66/25.72	53.39/24.04/25.76	52.40/23.58/24.83
LLaMA-13B	52.21/23.99/26.06	53.15/24.76/26.24	53.53/24.67/26.83	53.06/23.84/26.06
Vicuna-7B	55.01/25.26/26.42	54.32/25.32/26.24	54.71/25.27/26.15	54.64/24.94/26.07
Vicuna-13B	55.07/26.18/26.78	55.30/25.96/26.98	56.31/26.45/**27.02**	54.87/25.14/26.17
LongT5	56.29/26.67/24.85	55.92/26.57/25.14	55.80/26.39/25.21	55.40/26.10/24.43
LED-Base	55.56/25.47/23.04	53.98/24.66/23.36	55.89/24.72/24.03	55.05/25.51/24.19
Legal-LED	56.10/25.50/23.74	55.46/25.82/24.20	56.12/25.36/24.85	55.88/26.10/24.13
LED-Large	56.43/26.49/24.92	56.69/**27.04**/25.37	**57.15**/26.44/25.52	55.75/25.85/24.41
Average improvement %		−0.30/**0.49**/0.66	**1.06**/0.29/**1.76**	−0.27/−0.84/−0.73
CLSum-UK				
LLaMA-7B	60.68/27.65/26.04	60.83/28.29/26.68	60.47/27.14/25.90	60.37/27.37/26.05
LLaMA-13B	61.13/28.49/26.52	60.81/28.21/26.29	60.69/28.44/26.75	61.27/28.99/27.12
Vicuna-7B	61.42/29.04/26.83	61.58/29.28/27.10	61.63/29.46/27.30	61.53/28.78/26.77
Vicuna-13B	61.47/29.15/27.07	61.48/29.37/27.38	61.27/28.71/27.16	61.44/29.24/27.00
LongT5	59.62/28.08/26.64	59.76/27.81/26.22	60.27/28.62/26.86	60.54/29.23/26.93
LED-Base	62.18/28.92/25.91	**62.63**/29.52/26.58	62.41/30.81/27.41	61.61/28.49/26.02
Legal-LED	62.59/29.37/25.93	62.43/29.51/26.63	62.43/30.68/27.57	62.05/28.67/25.94
LED-Large	61.55/29.09/26.27	61.38/28.80/26.64	62.50/**31.16**/**28.17**	61.98/29.41/26.81
Average improvement %		0.05/0.44/1.11	**0.21**/**2.24**/**2.82**	0.04/0.19/0.68
CLSum-AUS				
LLaMA-7B	56.24/28.56/28.71	55.87/28.92/29.38	57.27/26.07/26.39	56.78/28.93/29.29
LLaMA-13B	58.76/31.04/30.64	58.69/31.03/30.70	57.81/30.02/29.74	58.25/31.10/30.73
Vicuna-7B	57.95/30.46/30.77	58.18/27.92/27.64	57.84/30.10/30.70	57.17/29.88/30.33
Vicuna-13B	58.17/30.51/30.86	58.10/30.71/30.98	57.17/30.30/31.27	58.57/31.22/30.96
LongT5	61.99/31.82/31.55	60.90/30.51/30.31	61.57/31.52/31.18	60.89/30.77/30.74
LED-Base	62.65/32.38/30.92	61.75/31.63/30.52	**63.14**/**35.19**/**33.32**	62.40/32.14/30.84
Legal-LED	62.42/32.11/30.54	61.61/31.70/30.47	62.77/34.44/32.61	62.27/31.88/30.78
LED-Large	62.64/32.78/31.57	62.44/32.38/31.14	62.72/32.66/31.27	63.07/33.01/31.47
Average improvement %		−0.66/−1.92/−1.76	**−0.11**/**0.09**/**0.31**	−0.29/−0.27/−0.14

assist LLMs in achieving commendable results by fine-tuning on a small set of labeled samples. Our experiment results verify the effectiveness of SFT.

We discover that GPT-3.5-turbo[16] fine-tuned with RLHF (OpenAI, 2022) has difficulty in generating case statements. The RLHF process trains the model to avoid generating illegal content. However, court judgment documents usually require an accurate statement of the parties' illegal facts, which are crucial bases for the judgment. RLHF used on general domains is not suitable for legal text generation. It requires a specially designed RLHF process to adequately cater to the complex requirements in the legal field. Court judgment summarization models' outputs should accurately and objectively reflect the cases' facts and the court's decisions. There should be no factual or logical errors. Parties from different groups should be treated fairly.

7.4. Discussion on data augmentation methods

The performance of supervised models trained from scratch is typically constrained by the training set size. As introduced in Section 5.1, we adopt and compare different data augmentation methods, including rephrasing, knowledge-constrained rephrasing, and back translation, to expand the training sets and reduce overfitting to the limited training samples. In our experiments, we doubled the training set size using each data augmentation method. Table 7 shows the impact of three data augmentation methods on summarization results. These data augmentation methods bring different performance gains to summarization models trained on

---

[16] We adopt the GPT-3.5-turbo-0301 API from Azure Cloud.

Table 8
Effect of the amount of trainable parameters in the QLoRA adapter. R1, R2, and RL are the F_1 score of ROUGE-1, ROUGE-2, and ROUGE-L.

Dataset	Method	Rank = 8 R1/R2/RL	Rank = 16 R1/R2/RL	Rank = 32 R1/R2/RL
CLSum-CA	LLaMA-7B	47.91/18.10/20.74	40.35/13.40/20.93	42.04/13.09/21.49
	LLaMA-13B	48.09/17.00/20.45	48.54/14.42/19.16	46.86/16.50/20.91
	Vicuna-7B	47.62/17.36/22.05	48.26/17.47/22.49	46.22/16.13/22.39
	Vicuna-13B	**50.66**/**19.22**/22.68	49.66/19.04/22.29	50.44/19.14/**22.84**
CLSum-HK	LLaMA-7B	51.71/23.30/26.18	53.73/24.31/26.32	53.64/24.25/26.16
	LLaMA-13B	52.21/23.99/26.06	54.13/25.31/26.61	54.89/25.83/26.90
	Vicuna-7B	55.01/25.26/26.42	55.02/25.20/26.40	55.13/25.82/26.58
	Vicuna-13B	55.07/26.18/26.78	**55.82**/**27.25**/**27.54**	**55.82**/26.95/27.00
CLSum-UK	LLaMA-7B	60.68/27.65/26.04	61.04/27.88/26.05	60.55/27.96/26.21
	LLaMA-13B	61.13/28.49/26.52	60.75/28.19/26.34	60.52/28.14/26.03
	Vicuna-7B	61.42/29.04/26.83	61.45/28.44/26.56	61.19/28.80/26.47
	Vicuna-13B	**61.47**/**29.15**/**27.07**	61.30/28.42/26.71	60.93/28.66/26.81
CLSum-AUS	LLaMA-7B	56.24/28.56/28.71	56.21/28.61/28.95	56.32/28.55/28.82
	LLaMA-13B	58.76/31.04/30.64	**59.15**/**31.82**/**31.81**	58.62/31.35/31.60
	Vicuna-7B	57.95/30.46/30.77	56.98/30.14/30.35	57.45/30.34/31.00
	Vicuna-13B	58.17/30.51/30.86	57.19/30.09/30.94	57.88/30.56/30.94

different subsets of CLSum. Experimental results verify that our proposed knowledge-constrained rephrasing method is helpful in the absence of labeled data. As shown in Table 2, CLSum-CA has the smallest training set. Data augmentation methods bring the most significant performance gain to summarization models trained on this subset. CLSum-AUS has the largest training set. Data augmentation methods bring marginal performance gain to models trained on that subset. This verifies that our data augmentation method primarily mitigates the impact of insufficient labeled data. The original rephrasing method and back translation method can benefit the ROUGE-L scores, but they often yield negative effects on ROUGE-1 and ROUGE-2 scores. Without the constraints of legal knowledge, there may be many errors in the data synthesized by data augmentation, which can adversely affect the summarization performance. Adding constraints in the rephrasing process can ensure the accurate use of legal terms in the synthesized data. This helps train models to accurately use relevant terms when generating judgment summaries.

7.5. Discussion on adapters' trainable parameters

The adapter is a small set of trainable parameters added to the large language models. We use the Low-rank Adapter (LoRA) (Dettmers et al., 2023; Hu et al., 2021) to reduce the consumption of GPU memory when fine-tuning LLaMA and Vicuna models. As shown in Eq. (4), LoRA supplements the original linear projection $h = W_0 x$ with an additional factorized projection. During training, $W_0 \in \mathbb{R}^{d \times k}$ remain unchanged, whereas $A \in \mathbb{R}^{r \times k}$ and $B \in \mathbb{R}^{d \times r}$, which have a rank $r \ll \min(d, k)$, comprise trainable parameters.

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{4}$$

Table 8 shows the impact of LoRA's rank r on summarization results. The number of trainable parameters in the adapters expands as the rank r increases. Results show that increasing the LoRA's rank r does not necessarily improve the generated summaries. The primary constraint on the model performance stems from the inadequate quantity of training samples.

7.6. Case study

We conduct a case study to compare and analyze summaries generated by different models. Fig. 9 presents fragments in the target summary and different models' output summaries. When comparing these summaries, we find that summaries produced by extractive summarization methods lack fluency and readability. The summaries generated by LLM (Vicuna) under the zero-shot setting are quite different from the target summary in format. The summaries generated by the fine-tuned models (LED and Vicuna) are closer to the target summaries in both content and format, even though the models used here were fine-tuned with only 100 samples. This case study demonstrates that supervised fine-tuning is crucial for ensuring that the generated summaries meet the content and formatting requirements.

8. Conclusion and future work

This paper introduces CLSum, a large-scale summarization dataset covering court judgments from four common law jurisdictions, including Canada, Australia, the United Kingdom, and Hong Kong SAR. Besides, we propose a large language model (LLM) based solution for the low-resource court judgment summarization. We present a series of methods to deal with three challenges: (1) training supervised summarization models with very limited labeled data, (2) identifying the salient content dispersed within the long judgment document, and (3) improving the efficiency of summarization models and training methods to process long input

---

Output Summary

Target Summary: 1. On 11 August 2010, the Respondent, Mr Ng (the victim) and his friend Mr Leung were involved in a brawl during which the Respondent punched Mr Ng on the bridge of his nose, causing him to fall face upward on the ground. The Respondent then kicked Mr Ng several times, fracturing Mr Ng's ribs. Mr Ng died the following day. The cause of death was traumatic head injury. In other words, the Respondent's punch, but not his kicks, caused Mr Ng's death. 2. The Respondent was charged with murder. He claimed to be acting in self-defence but was found guilty of murder by the jury. The Judge had directed the jury on the basis that the intention necessary for murder could be deduced from the Respondent's action in punching Mr Ng followed by the kicks. The Court of Appeal allowed the Respondent's appeal, setting aside the conviction for murder and substituting a conviction for manslaughter on the basis of unlawful act. It held that the Judge should have reminded the jury that the Respondent's kicks to Mr Ng did not kill him and directed the jury to consider whether the punch, which did kill him, was inflicted with the intention to cause grievous bodily harm. The Appellant sought to set aside the Court of Appeal's judgment and to have substituted a conviction for manslaughter on the basis of diminished responsibility relying on fresh medical evidence as to the Respondent's mental condition at the time of the attack. 3. The question of law was how, in cases of violence, the jury should be directed in inferring the necessary intent for the offence charged. Should the jury be directed to focus only on the blow or blows causative of the prohibited outcome or should they instead be directed to focus on all the circumstances, including all the blows struck or other matters capable of demonstrating the defendant's intent? 4. The Court noted that the Respondent's assault occupied a short period of time, occurred at the same physical location, was without any interruption and was unaccompanied by any statement indicating a change of intention. His actions were all part and parcel of a single attack. The Court held that in cases of violence involving a single assault consisting of multiple blows, the jury should be directed in inferring the necessary intent to focus on all the circumstances, including all the blows struck or other matters capable of demonstrating the defendant's intent. ...

TextRank: The question of law raised in this appeal is how, in cases of violence, the jury should be directed in inferring the necessary intent for the offence charged. In the context of this particular case in which the respondent was charged with murder (the facts of which are summarised below), the Court of Appeal[1] was of the former view, holding that the jury should be directed to focus only on the respondent's intention when throwing the initial punch which ultimately caused the victim's death. The Judge[2], on the other hand, had directed the jury at the trial on the basis of the latter view, inviting the jury to consider all the circumstances including the kicks administered by the respondent to the victim after the initial punch. There is now no doubt that the respondent was seriously mentally disturbed at the time of the offence (a point to which it will be necessary to return later in this judgment) but, although he had been prepared to plead guilty to unlawful act manslaughter (which plea the prosecution declined to accept), the defence did not proffer a plea to manslaughter by reason of diminished responsibility. On the issue of intention to commit murder, the Judge directed the jury on the basis that they could deduce an intention to cause really serious bodily harm from the respondent's action in pushing or punching the victim followed by the substantial kicks which fractured his ribs and that both these actions were administered with significant force. ...

LED_large finetuned: 1. On 11 August 2010, the Respondent and two other men, Mr Ng Fuk Keung (the victim) and Mr Leung Kam Shing (the other man) were involved in a brawl during which the respondent punched the victim on the bridge of the nose, causing him to fall to the ground on his back. While the victim was lying on the ground unconscious and bleeding from his head, the respondent kicked him three or four times in the ribs with great force. 2. The Respondent then threw a second punch at Mr. Leung, which although he dodged caught him on his left hand and fractured his left ring finger. After that, he kicked the deceased on his waist three to four times, fracturing his ribs. The victim died the following day, 12 August 2010. The autopsy report gave the cause of death as traumatic head injury. 3. The question of law raised in this appeal was how, in cases of violence, the jury should be directed in inferring the necessary intent for the offence charged. Should the jury be directed to focus only on the blow or blows causative of the prohibited outcome or should they instead be directed instead to focus on all the circumstances, including all the blows struck or other matters capable of demonstrating the defendant's intent? 4. In the present case, the Court of First Instance was of the view that the trial judge should have directed the jury at the trial on the basis of the latter view, directing the jury to consider all of the circumstances including the kicks administered by the respondent to the victim after the initial punch. ...

Vicuna 13B zero-shot: The article discusses a case in Hong Kong where a man was charged with murder for throwing a punch that led to the death of another man. The Court of Appeal overturned the conviction for murder and substituted it with a conviction for unlawful act manslaughter. The issue at hand was how the jury should be directed in inferring the necessary intent for the offence charged in cases of violence. The Judge directed the jury to focus only on the blow or blows causative of the prohibited outcome, while the Court of Appeal held that the The article discusses a case in which a man was convicted of murder for throwing a single punch that killed another man. The Court of Appeal overturned the murder conviction and substituted it with a conviction for unlawful act manslaughter. The Court of Appeal considered whether the punch and subsequent kicks were part of the same transaction or separate acts of violence. The prosecution argued that the incident was one continuous transaction, while the defense argued that the focus should be on the particular act that caused the death. The Court ultimately allowed the appeal The article discusses the circumstances under which a conviction for murder can be returned in cases where the prosecution can prove that some acts were done with the necessary intent but can not prove that other acts were done with that intent. It is an established general proposition that a jury should be directed to take into account all of the circumstances in order to determine what inferences may be drawn regarding a defendant's intention in relation to a particular act. The proposition that all the circumstances are relevant to the question of drawing inferences as to The article discusses the case of R v Chan, where the defendant was convicted of murder for punching the victim in the face, which led to the victim's death. The Court of Appeal quashed the conviction for murder and substituted it with a conviction for manslaughter ...

Vicuna 13B finetuned: 1. The issue in this appeal was how, in cases of violence, the jury should be directed in inferring the necessary intent for the offence charged. Should the jury be directed to focus only on the blow or blows causative of the prohibited outcome or should they instead be directed to focus on all the circumstances, including all the blows struck or other matters capable of demonstrating the defendant's intent? 2. The Court of Final Appeal held that the jury should be directed to focus on all the circumstances, including all the blows struck or other matters capable of demonstrating the defendant's intent. The respondent was convicted of murder and sentenced to life imprisonment. The Court of Appeal allowed the respondent's appeal and substituted a conviction for unlawful act manslaughter in place of the murder conviction. The Court of Final Appeal allowed the prosecution's appeal against the Court of Appeal's judgment and substituted a conviction for manslaughter by reason of diminished responsibility in place of the unlawful act manslaughter conviction. The Court of Final Appeal held that the trial judge was correct in directing the jury to take into account all of the circumstances in order to determine what inferences may be drawn regarding the appellant's intention in relation to the act that caused the victim's death. The Court of Final Appeal held that the trial judge was correct in directing the jury to take into account all of the circumstances in order to determine what inferences may be drawn regarding the appellant's intention in relation to the act that caused the victim's death. The respondent punched the victim in the face, lashed out at Mr Leung with his fist (fortunately not connecting with anything other than Mr Leung's hand), and then immediately thereafter viciously kicked the victim as he lay on the ground. This was all part and parcel of a single attack. ...

Fig. 9. The output summaries of an example from the CLSum-HK.

documents and summaries. This is the first court judgment summarization work adopting LLMs in data augmentation, summary generation, and evaluation. Specifically, we design an LLM-based data augmentation method named knowledge-constrained

rephrasing. It introduces legal knowledge into the prompts of LLMs to constrain the synthesized text to accurately use legal concepts. Furthermore, we design a legal knowledge enhanced evaluation metric named LTScore to assess the generated legal text. We employ various summarization methods as baselines and benchmark them on our CLSum dataset. Our experimental results verify that the LLM-based summarization methods can perform well in the few-shot and zero-shot settings. Our LLM-based data augmentation method can alleviate the impact of low data resources. We carry out comprehensive comparative experiments to find essential model components and settings that are capable of enhancing summarization performance, including the training set size, foundation model architecture, SFT, and the RLHF process.

In future work, we aim to improve the dataset, summarization models, and evaluation metrics. We plan to build a larger dataset covering judgments from more jurisdictions. To further improve summarization models, we plan to use larger LLMs and design specific alignment mechanisms that adequately cater to the complex requirements in the legal field. We also plan to design more sophisticated evaluation methods to accurately evaluate the generated legal text. Overall, low-resource court judgment summarization remains an open problem, which offers the potential for improvement and requires novel solutions to address the associated issues.

CRediT authorship contribution statement

Shuaiqi Liu: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jiannong Cao:** Writing – review & editing, Supervision, Funding acquisition. **Yicong Li:** Writing – review & editing, Validation, Investigation, Data curation. **Ruosong Yang:** Writing – review & editing, Investigation. **Zhiyuan Wen:** Writing – review & editing, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

Appendix

A.1. Summarization models' few-shot performance

Table 9 presents various summarization models' few-shot performance. We fine-tune summarization models on training sets of increasing size (from ten examples to hundreds of examples). Figs. 5, 6, and 7 visualize the impact of training set size on the ROUGE-2 scores, BARTScore, and LTScore of the generated summaries. Figs. 10 and 11 visualize the impact of training set size on the generated summaries' LTScore-LED and LTScore-Vicuna. These LTScore are calculated by single foundation model (LED or Vicuna). Their Pearson correlation with other evaluation metrics is shown in Fig. 8. For relevant result analysis, please refer to Section 7.1.

A.2. Summarization models' details

These summarization models' implementation details are shown in Table 10.

Table 9

Automatic evaluation results of summarization models' few-shot performance. "N examples" denotes using N examples when fine-tuning models. R1, R2, and RL are the F_1 score of ROUGE-1, ROUGE-2, and ROUGE-L.

Method	CLSum-CA			
	10 examples R1/R2/RL	50 examples R1/R2/RL	100 examples R1/R2/RL	500 examples R1/R2/RL
LLaMA$_{7B}$	46.41/15.95/19.00	45.61/15.80/20.08	47.91/18.10/20.74	–
LLaMA$_{13B}$	44.08/16.80/18.30	45.44/15.70/20.22	48.09/17.00/20.45	–
Vicuna$_{7B}$	47.94/16.78/20.77	47.02/17.00/21.64	47.62/17.36/22.05	–
Vicuna$_{13B}$	48.36/**17.50**/20.32	49.78/19.29/**22.72**	50.66/19.22/**22.68**	–
LongT5	48.97/12.79/18.10	52.77/19.23/21.15	55.85/19.98/21.48	–
LED$_{Base}$	51.41/16.76/20.75	55.10/19.39/21.36	54.57/19.63/21.32	–
Legal-LED	51.94/16.65/20.80	54.63/19.10/21.43	56.04/20.33/21.73	–
LED$_{Large}$	**54.37**/17.29/**20.85**	**56.27**/**19.52**/21.54	**57.23**/**21.15**/22.65	–

Method	CLSum-HK			
	10 examples R1/R2/RL	50 examples R1/R2/RL	100 examples R1/R2/RL	500 examples R1/R2/RL
LLaMA$_{7B}$	53.15/23.15/24.31	50.66/22.81/25.39	51.71/23.30/26.18	–
LLaMA$_{13B}$	52.75/22.95/25.03	51.98/22.76/25.22	52.21/23.99/26.06	–
Vicuna$_{7B}$	**55.58**/**25.57**/**26.26**	54.84/25.26/26.50	55.01/25.26/26.42	–
Vicuna$_{13B}$	54.14/24.83/26.15	**56.04**/**26.99**/**27.67**	55.07/26.18/**26.78**	–
LongT5	51.35/19.38/21.39	55.36/24.81/23.50	56.29/24.67/24.85	–
LED$_{Base}$	53.03/21.36/21.89	53.62/23.52/22.65	55.56/25.47/23.04	–
Legal-LED	53.26/22.05/22.54	54.23/24.45/23.58	56.10/25.50/23.74	–
LED$_{Large}$	53.96/22.52/22.42	53.61/22.93/22.23	**56.43**/26.49/24.92	–

Method	CLSum-UK			
	10 examples R1/R2/RL	50 examples R1/R2/RL	100 examples R1/R2/RL	500 examples R1/R2/RL
LLaMA$_{7B}$	55.12/26.29/23.57	59.68/26.54/25.23	59.77/27.02/25.72	60.52/27.77/26.09
LLaMA$_{13B}$	60.70/27.38/25.47	59.61/26.37/25.51	60.83/**28.09**/26.26	61.26/28.54/**26.70**
Vicuna$_{7B}$	58.80/26.87/25.40	59.77/27.26/25.92	60.05/26.98/25.50	61.74/28.58/26.68
Vicuna$_{13B}$	60.00/**28.07**/**26.02**	**60.91**/**28.18**/**26.11**	60.42/28.01/**26.44**	61.05/28.65/26.46
LongT5	55.47/23.24/22.38	57.80/25.81/24.13	58.07/25.99/24.48	58.51/26.73/25.89
LED$_{Base}$	59.61/24.19/22.00	60.23/26.59/23.48	60.28/27.64/24.37	**62.06**/**29.40**/25.30
Legal-LED	**60.62**/25.56/23.45	60.71/26.55/23.78	**61.52**/27.59/24.49	61.62/28.97/25.67
LED$_{Large}$	59.33/24.28/22.55	60.78/26.79/24.05	61.42/27.78/24.58	61.78/28.90/26.28

Method	CLSum-AUS			
	10 examples R1/R2/RL	50 examples R1/R2/RL	100 examples R1/R2/RL	500 examples R1/R2/RL
LLaMA$_{7B}$	**58.07**/27.92/28.29	58.11/29.28/30.22	57.08/29.32/30.53	57.77/30.69/30.81
LLaMA$_{13B}$	57.73/27.61/27.85	57.37/**29.49**/**30.67**	59.19/**30.96**/**31.05**	59.32/31.60/**32.16**
Vicuna$_{7B}$	57.27/**29.65**/**29.20**	56.80/29.42/29.49	57.27/30.01/30.40	57.66/30.13/30.47
Vicuna$_{13B}$	55.03/27.45/27.27	55.70/28.58/29.52	56.41/29.12/29.95	57.73/30.41/30.95
LongT5	57.46/25.62/26.81	**60.02**/29.24/28.14	60.58/30.31/29.18	61.47/31.21/30.40
LED$_{Base}$	57.18/24.73/26.04	59.03/27.57/27.46	59.68/29.22/28.24	61.91/31.60/30.21
Legal-LED	57.97/25.72/26.63	59.84/28.85/28.16	60.00/29.37/28.52	61.86/31.74/30.24
LED$_{Large}$	56.73/24.25/26.41	59.29/27.81/28.42	**61.19**/29.90/29.22	**62.77**/**32.07**/31.05

Table 10

Details of summarization models.

Model	Architecture	Params	Enc/Dec layers	Heads	d_{model}	d_{ff}	Input Len
LED$_{base}$	Enc-Dec	161.8M	6	12	768	3072	16,384
LED$_{large}$	Enc-Dec	459.8M	12	16	1024	4096	16,384
Legal-LED	Enc-Dec	161.8M	6	12	768	3072	16,384
LongT5$_{base}$	Enc-Dec	247.6M	12	12	768	2048	16,384
BLOOM$_{560M}$	Dec Only	559M	24	16	1024	4096	2048
BLOOM$_{7B1}$	Dec Only	7.1B	30	32	4096	16,384	2048
LLaMA$_{7B}$	Dec Only	6.7B	32	32	4096	11,008	2048
LLaMA$_{13B}$	Dec Only	13.0B	40	40	5120	13,824	2048
Vicuna$_{7B}$	Dec Only	6.7B	32	32	4096	11,008	2048
Vicuna$_{13B}$	Dec Only	13.0B	40	40	5120	13,824	2048

(a) CLSum-CA

(b) CLSum-HK

(c) CLSum-UK

(d) CLSum-AUS
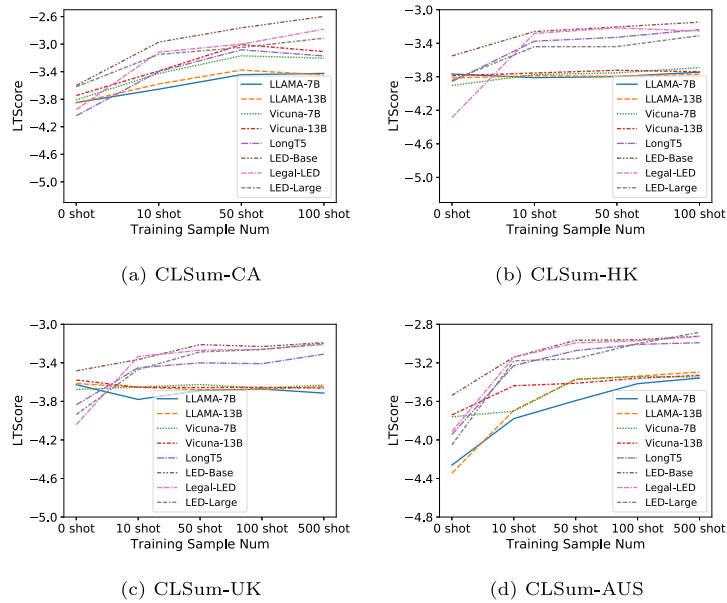
Fig. 10. Automatic evaluation result (LTScore$_{F_1}$-LED) on CLSum.

(a) CLSum-CA
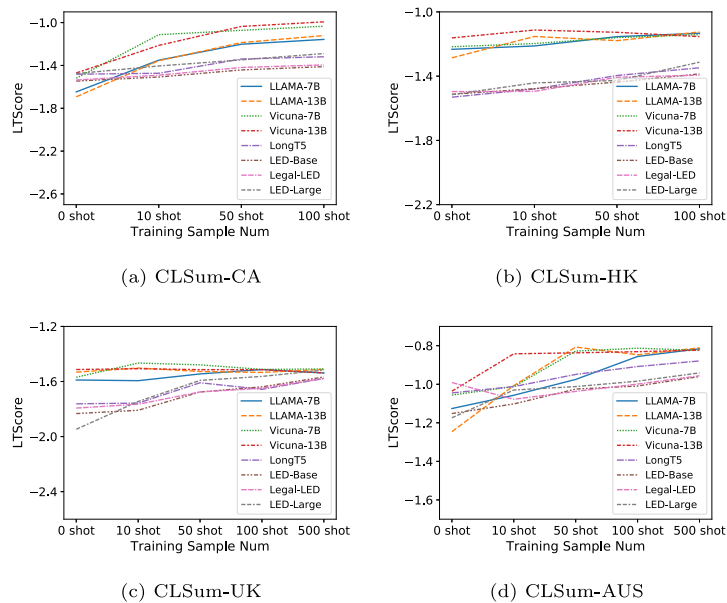
(b) CLSum-HK

(c) CLSum-UK

(d) CLSum-AUS

Fig. 11. Automatic evaluation result (LTScore$_{F_1}$-Vicuna) on CLSum.

References

Ashish, Bhan, & Mohit, Rohatgi Legal systems in India: Overview. URL uk.practicallaw.thomsonreuters.com/w-017-5278.

Bajaj, Ahsaas, Dangati, Pavitra, Krishna, Kalpesh, Kumar, Pradhiksha Ashok, Uppaal, Rheeya, Windsor, Bradford, et al. (2021). Long document summarization in a low resource setting using pretrained language models. In *Proc. ACL-sRW* (pp. 71–80).

Beltagy, Iz, Peters, Matthew E., & Cohan, Arman (2020). Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

Bhattacharya, Paheli, Hiware, Kaustubh, Rajgaria, Subham, Pochhi, Nilay, Ghosh, Kripabandhu, & Ghosh, Saptarshi (2019). A comparative study of summarization algorithms applied to legal case judgments. In *Proc. ECIR, part i 41* (pp. 413–428). Springer.

Chiang, Wei-Lin, Li, Zhuohan, Lin, Zi, Sheng, Ying, Wu, Zhanghao, Zhang, Hao, et al. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. lmsys.org/blog/2023-03-30-vicuna/.

Dai, Haixing, Liu, Zhengliang, Liao, Wenxiong, Huang, Xiaoke, Cao, Yihan, Wu, Zihao, et al. (2023). arXiv preprint arXiv:2302.13007.

Dao, Tri, Fu, Dan, Ermon, Stefano, Rudra, Atri, & Ré, Christopher (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. *Vol. 35*, In *Proc. NeurIPS* (pp. 16344–16359).

Dettmers, Tim, Lewis, Mike, Belkada, Younes, & Zettlemoyer, Luke (2022). Llm. int8 (): 8-bit matrix multiplication for transformers at scale. arXiv preprint arXiv:2208.07339.

Dettmers, Tim, Lewis, Mike, Shleifer, Sam, & Zettlemoyer, Luke (2022). 8-bit optimizers via block-wise quantization. In *Proc. ICLR*.

Dettmers, Tim, Pagnoni, Artidoro, Holtzman, Ari, & Zettlemoyer, Luke (2023). Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, & Toutanova, Kristina (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT, volume 1 (long and short papers)* (pp. 4171–4186).

Erkan, Günes, & Radev, Dragomir R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479.

Fabbri, Alexander, Li, Irene, She, Tianwei, Li, Suyi, & Radev, Dragomir (2019). Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proc. ACL* (pp. 1074–1084).

Farzindar, Atefeh, & Lapalme, Guy (2004). *LetSum, an automatic text summarization system in law field*. JURIX.

Feijó, Diego de Vargas, & Moreira, Viviane Pereira (2018). Rulingbr: A summarization dataset for legal texts. In *Proc. PROPOR* (pp. 255–264). Springer.

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

Fleiss, Joseph L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, *76*(5), 378.

Glaser, Ingo, Moser, Sebastian, & Matthes, Florian (2021). Summarization of german court rulings. In *Proc. NLLP* (pp. 180–189).

Grusky, Max, Naaman, Mor, & Artzi, Yoav (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proc. NAACL-HLT* (pp. 708–719).

Guo, Mandy, Ainslie, Joshua, Uthus, David, Ontanon, Santiago, Ni, Jianmo, Sung, Yun-Hsuan, et al. (2022). LongT5: Efficient text-to-text transformer for long sequences. In *Findings of NAACL* (pp. 724–736).

HKSAR DOJ The common law and rules of equity. URL www.doj.gov.hk/en/our_legal_system/the_common_law.html.

Hu, Edward J, Wallis, Phillip, Allen-Zhu, Zeyuan, Li, Yuanzhi, Wang, Shean, Wang, Lu, et al. (2021). LoRA: Low-rank adaptation of large language models. In *Proc. ICLR*.

Huang, Luyang, Cao, Shuyang, Parulian, Nikolaus, Ji, Heng, & Wang, Lu (2021). Efficient attentions for long document summarization. In *Proc. NAACL-HLT* (pp. 1419–1436).

Kanapala, Ambedkar, Pal, Sukomal, & Pamula, Rajendra (2019). Text summarization from legal documents: a survey. *Artificial Intelligence Review*, *51*, 371–402.

Kingma, Diederik P., & Ba, Jimmy (2015). Adam: A method for stochastic optimization. CoRR abs/1412.6980.

Lehman, Jeffrey S., & Phelps, Shirelle (2005). *West's Encyclopedia of American Law*: *vol. 6*, (p. 293).

Lewis, Mike, Liu, Yinhan, Goyal, Naman, Ghazvininejad, Marjan, Mohamed, Abdelrahman, Levy, Omer, et al. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. ACL* (pp. 7871–7880).

Li, Shuaimin, & Xu, Jungang (2023). MRC-sum: An MRC framework for extractive summarization of academic articles in natural sciences and medicine. *Information Processing & Management*, *60*(5), Article 103467.

Lin, Chin-Yew (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81).

Liu, Shuaiqi (2024). Neural abstractive summarization for long documents.

Liu, Shuaiqi, Cao, Jiannong, Deng, Zhongfen, Zhao, Wenting, Yang, Ruosong, Wen, Zhiyuan, et al. (2023). Neural abstractive summarization for long text and multiple tables. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, Shuaiqi, Cao, Jiannong, Yang, Ruosong, & Wen, Zhiyuan (2021). Highlight-transformer: Leveraging key phrase aware attention to improve abstractive multi-document summarization. In *Findings of ACL* (pp. 5021–5027).

Liu, Shuaiqi, Cao, Jiannong, Yang, Ruosong, & Wen, Zhiyuan (2022a). Generating a structured summary of numerous academic papers: Dataset and method. In *Proc. IJCAI* (pp. 4259–4265).

Liu, Shuaiqi, Cao, Jiannong, Yang, Ruosong, & Wen, Zhiyuan (2022b). Key phrase aware transformer for abstractive summarization. *Information Processing & Management*, *59*(3), Article 102913.

Liu, Shuaiqi, Cao, Jiannong, Yang, Ruosong, & Wen, Zhiyuan (2022c). Long text and multi-table summarization: Dataset and method. In *Findings of EMNLP 2022* (pp. 1995–2010).

Mihalcea, Rada, & Tarau, Paul (2004). Textrank: Bringing order into text. In *Proc. EMNLP* (pp. 404–411).

OpenAI (2022). Introducing ChatGPT. openai.com/blog/chatgpt.

Polsley, Seth, Jhunjhunwala, Pooja, & Huang, Ruihong (2016). Casesummarizer: a system for automated summarization of legal texts. In *Proc. COLING: system demonstrations* (pp. 258–262).

Raffel, Colin, Shazeer, Noam, Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140), 1–67.

Rajbhandari, Samyam, Rasley, Jeff, Ruwase, Olatunji, & He, Yuxiong (2020). Zero: Memory optimizations toward training trillion parameter models. In *Proc. SC* (pp. 1–16). IEEE.

Rehurek, R. (2010). Software framework for topic modelling with large corpora.. In *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks* (p. 45).

Salton, Gerard, & Buckley, Christopher (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.

Saravanan, Murali, Ravindran, Balaraman, & Raman, Shivani (2006). Improving legal document summarization using graphical models. *FAIA*, *152*, 51.

Sennrich, Rico, Haddow, Barry, & Birch, Alexandra (2016). Improving neural machine translation models with monolingual data. In *Proc. ACL* (pp. 86–96).

Shazeer, Noam, & Stern, Mitchell (2018). Adafactor: Adaptive learning rates with sublinear memory cost. In *Proc. ICML* (pp. 4596–4604). PMLR.

Shukla, Abhay, Bhattacharya, Paheli, Poddar, Soham, Mukherjee, Rajdeep, Ghosh, Kripabandhu, Goyal, Pawan, et al. (2022). Legal case document summarization: Extractive and abstractive methods and their evaluation. In *Proc. AACL-IJCNLP* (pp. 1048–1064).

Touvron, Hugo, Lavril, Thibaut, Izacard, Gautier, Martinet, Xavier, Lachaux, Marie-Anne, Lacroix, Timothée, et al. (2023). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, et al. (2017). Attention is all you need. In *Proc. neurIPS* (pp. 5998–6008).

Wang, Xun, Nishino, Masaaki, Hirao, Tsutomu, Sudoh, Katsuhito, & Nagata, Masaaki (2016). Exploring text links for coherent multi-document summarization. In *Proc. COLING: technical papers* (pp. 213–223).

Wolf, Thomas, Chaumond, Julien, Debut, Lysandre, et al. (2020). Transformers: State-of-the-art natural language processing. In *EMNLP* (pp. 38–45).

Workshop, BigScience, Scao, Teven Le, Fan, Angela, Akiki, Christopher, Pavlick, Ellie, Ilić, Suzana, et al. (2022). Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100.

Wu, Xiaoxia, Li, Cheng, Aminabadi, Reza Yazdani, Yao, Zhewei, & He, Yuxiong (2023). Understanding INT4 quantization for transformer models: Latency speedup, composability, and failure cases. arXiv preprint arXiv:2301.12017.

Yao, Jin-ge, Wan, Xiaojun, & Xiao, Jianguo (2017). Recent advances in document summarization. *Knowledge and Information Systems*, *53*(2), 297–336.

Yuan, Weizhe, Neubig, Graham, & Liu, Pengfei (2021). Bartscore: Evaluating generated text as text generation. *Vol. 34*, In *Proc. NeurIPS* (pp. 27263–27277).

Zhang, Jingqing, Zhao, Yao, Saleh, Mohammad, & Liu, Peter (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proc. ICML* (pp. 11328–11339). PMLR.

Zhuang, Li, Jing, Feng, & Zhu, Xiao-Yan (2006). Movie review mining and summarization. In *Proc. CIKM* (pp. 43–50).