SKGSum: Structured Knowledge-Guided Document Summarization

Qiqi Wang[1†], **Ruofan Wang**[1†], **Kaiqi Zhao**[1‡*], **Robert Amor**[1‡], **Benjamin Liu**[2‡], **Jiamou Liu**[1‡], **Xianda Zheng**[1†], **Zijian Huang**[1†]
[1]School of Computer Science, Faculty of Science
[2]Department of Commercial Law, Faculty of Business and Economics
University of Auckland, New Zealand
[†]{qwan857,rwan551,xzhe162,zhua764}@aucklanduni.ac.nz,
[‡]{kaiqi.zhao,trebor,b.liu,jiamou.liu}@auckland.ac.nz

Abstract

A summary structure is inherent to certain types of texts according to the Genre Theory of Linguistics. Such structures aid readers in efficiently locating information within summaries. However, most existing automatic summarization methods overlook the importance of summary structure, resulting in summaries that emphasize the most prominent information while omitting essential details from other sections. While a few summarizers recognize the importance of summary structure, they rely heavily on the predefined labels of summary structures in the source document and ground truth summaries. To address these shortcomings, we developed a Structured Knowledge-Guided Summarization (SKGSum) and its variant, SKGSum-W, which do not require structure labels. Instead, these methods rely on a set of automatically extracted summary points to generate summaries. We evaluate the proposed methods using three real-world datasets. The results indicate that our methods not only improve the quality of summaries, in terms of ROUGE and BERTScore, but also broaden the types of documents that can be effectively summarized.[1]

1 Introduction

Automatic summarization is one of the most effective solutions to help people read lengthy content quickly and comprehend key information that interests them. Furthermore, well-structured summaries can significantly improve the accessibility of information by offering organized key points.

In linguistics, the notion that texts of the same type display similar structural features is widely accepted in *Discourse Structure* (Brown and Yule, 1983; Choubey et al., 2020) and *Genre Theory* (Swales and Swales, 1990; Martin, 1992). Texts of the same genre share similar structures to meet specific communicative objectives. These structures have evolved based on cultural norms and the shared expectations of both the writer and reader (Swales and Swales, 1990).

Building on Genre Theory, numerous educational research studies strongly advise students or new writers to write structurally articulated summaries to enhance their readability, such as those related to news articles (Yang, 2015; Whiting et al., 2018), government reports (Keepnews, 2016), legal documents (Makdisi and Makdisi, 2009; Kurzon, 1985), medical reports (Yuan ke and Hoey, 2014), and research articles (Hill, 1991). Typically, a structured summary comprises several summary points, each representing a specific piece of information from the source document. We find that various open-source long document summaries exhibit certain structures (Hill, 1991; Yuan ke and Hoey, 2014; Whiting et al., 2018; Yang, 2015), as elaborated in Section 2.1.

However, most existing text summarization methods (Liu et al., 2022; Ravaut et al., 2022; Xu et al., 2022a; Yao et al., 2023; Phang et al., 2023) neglect *summary structures*. While many researchers agree that structured information is essential to summarization, most of the current models focus on the *source document structures* or the relations between source sentences instead of the structure of summaries (Balachandran et al., 2021; Xiao and Carenini, 2019; Cao and Wang, 2022; Qiu and Cohen, 2022). When implementing these models, a significant concern arises due to their tendency to overly emphasize certain sections while neglecting others. For instance, most news articles follow the inverted pyramid model under which the most crucial details (Introduction) are presented first, followed by supporting information (News Details), and then additional background or related commentary (Reactions). As observed in Table 1, when summarizing news articles from the Daily

*Kaiqi Zhao is the corresponding author.
[1]The code used in this paper will be released at https://github.com/77-qiqi-wang/SKGSum/.

Findings of the Association for Computational Linguistics: ACL 2024, pages 1857–1871

Table 1: Example summaries for news from Daily Mail, generated by abstractive (BART), knowledge-guided (GSum), large language models (GPT-4 and Llama2) summarizers, and the proposed method **SKGSum-W**. The colors Magenta, Cyan, and Brown represent the *Introduction*, *News Details*, and *Reaction* components, respectively. Sentences marked with ~~strikethrough~~ are contrary to the facts.

Human: Tong Shao, 20, was an international student from China attending Iowa State University. Her body was found in the trunk of her car in Iowa City on September 26. Police believe it had been for three weeks. She died of blunt force trauma and asphyxiation. Her boyfriend, Xiangnan Li, 23, was the last to see her, but flew to China on September 8, before Shao was officially missing. According to Tong's father, an arrest warrant has now been issued. However, Li has disappeared.
Abstractive (BART): Tong Shao, 20, was found dead in her Toyota Camry in Iowa City on September 26. Her boyfriend, Xiangnan Li, 23, was listed as a person of interest in the case. The two had met in China in 2011 and Li was the last to see her.
Guiding knowledge (GSum): Tong Shao, a chemical engineering student at Iowa State University, went missing in September 2014. After a three week search, police found her body stuffed in the trunk of her Toyota Camry in Iowa City. Xiangnan Li, 23, was listed as a person of interest in the case.
GPT-4: Tong Shao, a 20-year-old chemical engineering student at Iowa State University, was found murdered and stuffed in the trunk of her car in Iowa in 2014. Her boyfriend, Xiangnan Li, was the last person to see her alive and is considered a person of interest in the case. He bought a one-way ticket to China soon after Shao's disappearance.
Llama2: Tong Shao, a 20-year-old Chinese student at the University of Iowa, was found murdered in her car in Iowa City in September 2014.Her boyfriend, Xiangnan Li, a chemical engineering student at Iowa State University, went missing and an arrest warrant has been issued for him. Li is wanted for the first-degree murder of Tong. Tong's father, Chunsheng Shao, has pleaded with U.S. authorities to share evidence gathered in the case with Chinese authorities, as there is no extradition treaty between the two countries. ~~The case is ongoing, and police have not confirmed the arrest warrant.~~
SKGSum-W: Tong Shao, 20, was found dead in her Toyota Camry on September 26. Her body was found wrapped in a towel and in the trunk for three weeks. Xiangnan Li is believed to have been the last person to see Tong alive. He bought a one-way ticket to China in the days after her disappearance. Police have issued an arrest warrant for her boyfriend, Xiangnan Li, 23. The father of Tong Shao says he wants more to be done in solving her murder.

Mail[2], existing State-of-the-Art (SOTA) summarizers fall short in adequately covering the three crucial summary points: "Introduction", "News Details", and "Reactions", when compared to human-generated summaries. More specifically, abstractive summarizers omit information regarding "Reactions" and place lesser emphasis on "News Details". This is also observed in the knowledge-guided method (Dou et al., 2021), which similarly lacks coverage of "Reactions" and under-represents "News Details". Table 1 also shows that Large Language Models (LLMs) like GPT-4 lose "Reactions" information, while Llama2 overly concentrates on "Reactions", thereby losing details from the "News Details" section and introduces incorrect information. These discrepancies can be attributed to their primary focus on the most prominent information, leading to information loss.

The challenge in summarizing lengthy documents often arises from adopting an appropriate summary structure during the generation process. We argue that an effective document summarizer should generate **structured summaries** that cover diverse summary points, such as 'Introduction', 'News Details', and 'Reactions', which originate from the source documents. While some studies recognize the importance of summary structure, they require manual assignment of each sentence in the source document to a predefined summary structure label (Gidiotis and Tsoumakas, 2020; Elaraby and Litman, 2022). Alternatively, other research relies on human input to create summary structures and manually generate summary structure labels (Wang et al., 2023). Obviously, such a method is practical only for documents that have a clear, consistent, and fixed summary structure, such as legal documents.

To overcome the above challenge, we propose a Structured Knowledge-Guided Summarization method (**SKGSum**). **SKGSum** employs a set of key phrases, automatically extracted from human-written summaries (as detailed in Section 4.1), to represent each summary point. These key phrases are utilized to identify relevant words or sentences in the source document through a *summary point-document alignment layer*. This method enables **SKGSum** to effectively capture pertinent information for each summary point, thus addressing the issue of information loss. However, training **SKGSum** still requires summary structure labels. To remove the need for these labels during training and to broaden the application scope, we introduce a variant of **SKGSum**, named **SKGSum-W**. This variant synthesizes information from all summary points to generate a comprehensive summary.

Our contributions are summarized as follows: (1) We propose a Structured Knowledge-Guided Summarization (**SKGSum**) to identify essential information for each summary point from the source documents. (2) We propose **SKGSum-W**, a vari-

[2]The web link of the news: https://www.dailymail.co.uk/news/article-3026101/.

Table 2: Example Structured Summaries

Type	Example Summary	Summary Points
News	Survey of 1,000 firms showed half are less inclined to recruit obese people.	**Introduction**
	Believe they are 'lazy' and 'unable to fulfil their roles as required'. Comes after European court ruled obesity is a disability after 25st Danish childminder claimed he was sacked by local authority because he was fat.	**News Details**
	Specialist furniture such as larger chairs. Parking spaces next to the workplace. Dietary advice to overweight staff. Gym memberships. Opportunities to work from home.	**Reaction**
Government Report	Spent nuclear fuel-the used fuel removed from commercial nuclear power reactors is an extremely harmful substance if not managed properly. ...	**Background**
	GAO found that some organizations that oppose DOE have effectively used social media to promote their agendas to the public, but had no coordinated outreach strategy.	**Findings**
	GAO is making no new recommendations.	**Recommendation**
New Zealand Judgment	Successful application by g for vest order.	**Decision**
	G seek order vest trust property in himself and a trustee; original trustee have lose capacity.	**Fact**
	Hold, vest order make a sought.	**Reason**

ant of **SKGSum** that does not require summary structure labels. (3) Experiments on three open datasets show that **SKGSum** improves the existing summarizers by up to 7.05% and 1% in terms of ROUGE scores and BERTScores, respectively.

2 Motivation and Problem Formulation

2.1 Motivation

As mentioned in Section 1, high-quality summary is always structured. In this regard, we analyzed the summaries of three real-world datasets, namely CNN and Daily Mail (CNN/DM) News Articles (Hermann et al., 2015; Nallapati et al., 2016), Government Reports (Huang et al., 2021), and New Zealand Judgments (Wang et al., 2023), and observed that a common summary structure exists for each type of document. Table 2 shows some examples of our findings.

Previous studies, however, either overlook this aspect or require human-predefined structure labels. Motivated by this gap, we propose utilizing summary points as guiding information to generate structured summaries. This approach aims to harness the inherent structure in human-written summaries, applying it more broadly across various types of documents.

2.2 Problem Definition

Our approach aims to generate structured summaries for lengthy documents. Specifically, let $\mathcal{D}_{\text{train}} = \{(D_1, S_1), (D_2, S_2), \ldots, (D_N, S_N)\}$ be a dataset comprising N documents and their corresponding summaries, where D_i and S_i represent the i-th source document and its human-written summary (also known as the gold summary), respectively. Each document is represented as a set of words, i.e., $D_i = \{w_{i1}, \ldots, w_{i|D_i|}\}$.

The gold summary S_i for the i-th document is partitioned into p gold summary parts according to p summary points, i.e., $S_i = \{S_i^1, S_i^2, \ldots, S_i^p\}$, where S_i^k is the summary of the k-th summary point for document D_i, represented as a set of m words $S_i^k = \{s_{i1}^k, \ldots, s_{im}^k\}$. The learning objective of structured summarization is to train a text summarizer M to generate a summary $G_i = \{G_i^1, \ldots, G_i^p\}$ of p summary points for each document D_i such that the sum of point-wise differences between the gold summaries and the generated summaries, i.e., $\sum_{i=1}^{N} \sum_{k=1}^{p} d(S_i^k, G_i^k)$, is minimized. Here, $d(\cdot, \cdot)$ is a distance function.

3 Methodology

Most summarizers employ a single document encoder and a single decoder, as illustrated at the top of Figure 1. Such an architecture cannot generate content regarding different summary points. Alternatively, one can train separate decoders to decode different parts of a summary as illustrated in the second part of Figure 1. To address the limitations of previous works, we propose a Structured Knowledge-Guided Summarization method (**SKGSum**) in Section 3.1. We further discuss a variant of **SKGSum**, named **SKGSum-W**, for datasets without gold part summaries in Section 3.2.

3.1 Structured Knowledge-Guided Summarizer

Drawing inspiration from the human writing process, **SKGSum** employs summary points as guiding information to find the content relevant to each summary point, as illustrated in part 3 of Figure 1.

The principal innovation of **SKGSum** lies in the *Structure-Guided Document Encoder* (Sec-

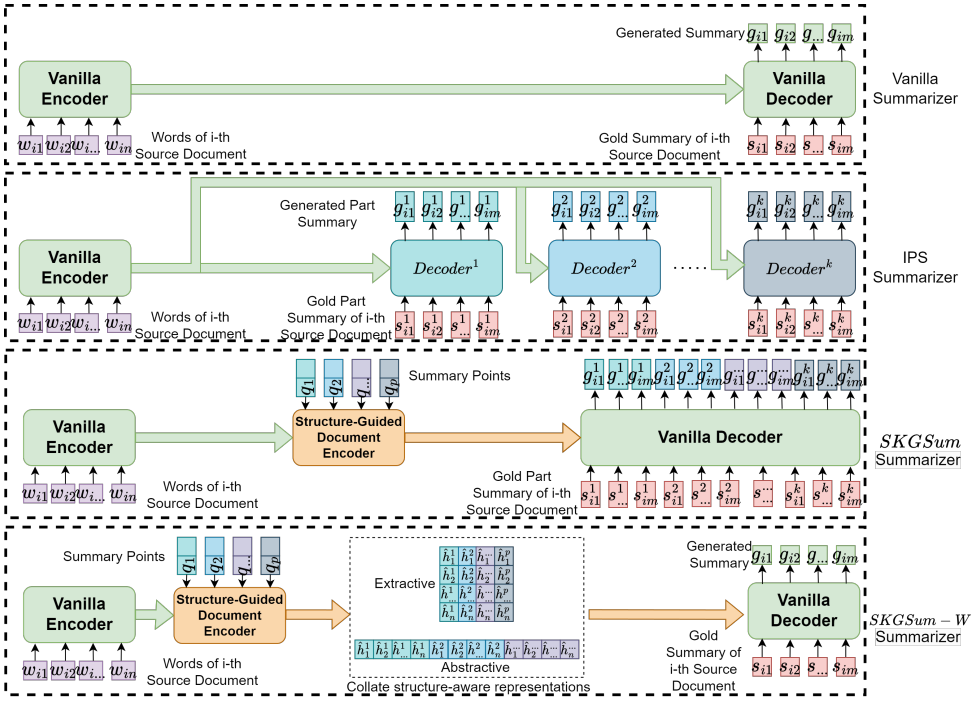

Figure 1: Summarizer architectures in Training Process. From top to bottom are (1) a vanilla summarizer with the Encoder-Decoder framework; (2) **IPS** (Wang et al., 2023); (3) **SKGSum**; and (4) **SKGSum-W**.

tion 3.1.2), which refines the encoder's output to generate summary point-specific representations for the sentences or words in the source documents. These representations are subsequently utilized as input for the decoder to generate summaries specific to each summary point.

3.1.1 Document Encoder and Decoder

We design **SKGSum** with a flexible architecture that can accommodate various document encoders and decoders, catering to different scenarios, i.e., extractive and abstractive summarization. Formally, a document encoder converts a source document $D = \{w_1, w_2, ..., w_{|D|}\}$ to a sequence of d-dimensional latent vectors $h_1, h_2, ..., h_n \in \mathbb{R}^d$. For extractive methods, each latent vector corresponds to a sentence in the source documents ($n << |D|$). For abstractive methods, each latent vector corresponds to a word ($n = |D|$). The document decoder generates a sequence of predictions $\{g_1, g_2, ..., g_m\}$ from the document vector representation $\{h_1, h_2, ..., h_n\}$.

In this paper, we adopt the encoders and decoders in ExtSum-LG (Xiao and Carenini, 2019) and BART (Lewis et al., 2020) for extractive and abstractive summarization, respectively.

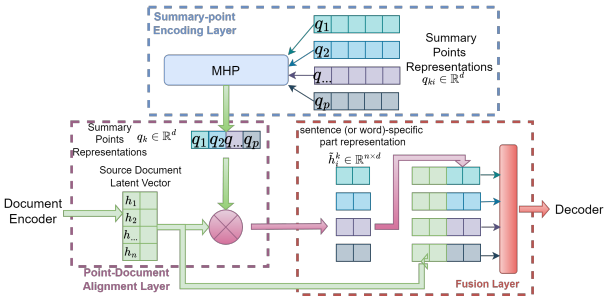

Figure 2: The structure-guided document encoder architecture of **SKGSum** model.

3.1.2 Structure-guided document encoder

Figure 2 demonstrates the architecture of our proposed structure-guided document encoder. Given the latent vector representations of sentences or words, i.e., $\{h_1, ..., h_n\}$ and p summary points as inputs, the structure-guided document encoder outputs the source document representation $\{\hat{h}_1^k, ..., \hat{h}_n^k \in \mathbb{R}^d\}$ specific to the k-th summary point. These summary point-specific document representations will then be fed to the decoder to generate summaries regarding each summary point. The structure-guided document encoder contains three components: summary-point encoding, point-document alignment, and information fusion.

Summary-point Encoding Layer. We employ a set of key phrases pertinent to summary points

as guiding signals to identify relevant sentences or words for each part. Specifically, we encode each point into a latent vector representation, which is subsequently utilized in the point-document alignment component to align sentences or words from the source document with summary points. The summary-point encoding layer takes two steps.

In the first step, we extract a set of key phrases from ground truth summaries associated with the summary points (details are elaborated in Section 4.1). Given the key phrases, we employ the document encoder to initialize the embeddings of these key phrases. Formally, we denote the embedding of the i-th word for the k-th summary point as $p_{ki} \in \mathbb{R}^d$. It is noteworthy that these word embeddings will be updated during the training process.

In the second step, we utilize Multi-head Pooling (MHP) (Liu and Lapata, 2019a) to learn the summary-point representation $p_k \in \mathbb{R}^d$ of the k-th part from the word embeddings $p_{k1}, ..., p_{kn}$:

$$\begin{aligned} \alpha_{ij}^z &= (\mathbf{W}_z^P p_{ki})^T (\mathbf{W}_z^K p_{kj}) \\ head_z &= \sum_i \sum_j \frac{exp(\alpha_{ij}^z)}{\sum_{j'} exp(\alpha_{ij'}^z)} \mathbf{W}_z^V p_{kj} \\ p_k &= \mathbf{W}_o \left[head_1 \| head_2 \| ... \| head_r \right], \end{aligned} \tag{1}$$

where $\mathbf{W}_z^P \in \mathbb{R}^{d' \times d}, \mathbf{W}_z^K \in \mathbb{R}^{d' \times d}, \mathbf{W}_z^V \in \mathbb{R}^{d' \times d}$ are learnable weights for the z-th attention head, $\mathbf{W}_o \in \mathbb{R}^{d \times rd'}$ is a learnable weight matrix to aggregate the outcomes from the r attention heads. The MHP extracts and aggregates essential information from the word embeddings to get a fixed-length vector representation for each summary point.

Point-Document Alignment Layer. Some previous works fall short in considering the interrelations between different summary points (Elaraby and Litman, 2022; Wang et al., 2023). However, sentences in source documents can pertain to multiple sections of the summary (Butt, 2013; Banks-Smith, 2019). For example, sentences containing information about the news story can be crucial to both the "Introduction" and "News Details" parts.

To address this problem, we leverage summary points to identify relevant sentences or words from the source document. Specifically, we compute the semantic similarity between each sentence or word and the summary points, utilizing their latent vector representations. This approach allows us to determine a relevance score for each sentence or word in relation to each summary point. Then, we acquire a sentence- (or word-) specific summary point representation, denoted as $\tilde{h}_i^k$, by aligning the relevance score with the summary point representation:

$$\begin{aligned} \tilde{h}_i^k &= a_{ik} p_k, \\ a_{ik} &= \frac{exp(h_i^T p_k)}{\sum_{k'} exp(h_i^T p_{k'})}, \end{aligned} \tag{2}$$

where a_{ik} is the relatedness of the i-th sentence or word to the k-th summary point. The notation p_k denotes the k-th summary point representation obtained from Equation 1.

Information Fusion Layer. The information fusion layer combines the sentence- (or word-) specific summary point representation, i.e., $\tilde{h}_i^k$, and the latent representation of the sentence or word, i.e., h_i, to generate a unified representation for each sentence or word. The unified representation incorporates the information from the context of the words (or sentences) in the source document and their related summary point. The fusion layer can also incorporate additional knowledge $X_i \in \mathbb{R}^d$ for each word or sentence, such as section representations, sentence relations, previously selected sentences, etc. Specifically, the extra knowledge X_i can be incorporated by concatenating it with the corresponding $\tilde{h}_i^k$ and h_i. The information fusion layer is defined as:

$$\hat{h}_i^k = \mathbf{W}_f \left[h_i \| \tilde{h}_i^k \| X_i \right], \tag{3}$$

where $\mathbf{W}_f \in \mathbb{R}^{d \times 3d}$ represents a learnable weight matrix. The information fusion layer will output p representations for each sentence or word, each corresponding to a summary point. Then, the decoder utilizes the representations to generate the summary for the k-th summary point.

Following previous works (Xiao and Carenini, 2019), we also apply cross-entropy as loss functions. More details can be found in Appendix A.

3.2 Structured Knowledge-Guided Summarizer Without Label Requirement

Previous methodologies (Gidiotis and Tsoumakas, 2020; Elaraby and Litman, 2022; Wang et al., 2023) and **SKGSum** require structure labels of ground truth summaries for training. However, obtaining the structure labels demands significant human effort. To mitigate this limitation, we introduce a variation of **SKGSum** that eliminates the necessity for structure labels within each ground truth summary. We refer to this new variant as the

Structured Knowledge-Guided Summarizer without Label Requirement (**SKGSum-W**). **SKGSum-W** is designed to adapt to a wider range of summarization scenarios, offering flexibility in scenarios where structured labels are not available.

Specifically, **SKGSum-W** retains most settings from **SKGSum** and adds an additional step at the end of the Fusion Layer to integrate information specific to all summary points. The fusion layer outputs p structure-aware representation for each document, each corresponding to one of the summary points. To ensure compatibility with most existing summarizers, we provide two distinct designs suited for extractive and abstractive summarizers, respectively.

3.2.1 Extractive summarizer version

Extractive summarizers aim to determine each sentence in source documents into a binary class label (selected or unselected) to combine selected sentences as generated summaries. This means that the length of selected sentences is fixed. Thus, we concatenate the structure-aware representation $\hat{h}_i^k \in \mathbb{R}^{n \times d}$, obtained from Equation 3, by sentences i. The structure-aware representation for extractive methods is defined as $\hat{h} \in \mathbb{R}^{n \times pd}$:

$$\hat{h} = [\hat{h}_1^1 \| ... \| \hat{h}_1^p,\ \hat{h}_2^1 \| ... \| \hat{h}_2^p,\ ...,\ \hat{h}_n^1 \| ... \| \hat{h}_n^p]. \quad (4)$$

3.2.2 Abstractive summarizer version

The decoder of abstractive summarizers needs to process all information sequentially and always use a decoder with fixed-dimension (e.g., d-dimension) hidden states. Thus, we concatenate the structure-aware representation $\hat{h}_i^k \in \mathbb{R}^{n \times d}$, obtained from Equation 3, by summary points k. This results in a unified, structure-aware representation, denoted as $\hat{h} \in \mathbb{R}^{pn \times d}$, as illustrated in Equation 5. An added benefit of this process is that it enables the decoder to sequentially process information from each part of the source document.

$$\hat{h} = [\hat{h}_1^1 ... \hat{h}_n^1, \hat{h}_1^2 ... \hat{h}_n^2, ..., \hat{h}_1^p ... \hat{h}_n^p]. \quad (5)$$

We utilize the structured-guided document representations $\hat{h}$, obtained by Equations 4 and 5, as the input for the decoder. This strategic input enables the decoder to generate a complete summary informed by the summary points. Consequently, **SKGSum-W** can be trained on datasets with complete and undivided gold summary without the need for structure labels.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our proposed methods on the following open datasets: (1) *CNN/ DM News Articles*, a widely used dataset for summarization tasks, comprising over 300K news articles sourced from CNN and Daily Mail websites (Hermann et al., 2015; Nallapati et al., 2016), with an average length of 750 words; (2) *Government Reports*, a dataset from the U.S. Government Accountability Office website (*GAO*) encompasses 12,228 reports, with an average length of 9,409 words (Huang et al., 2021); (3) *New Zealand Judgment Dataset* (*NZJD*), a law dataset that includes 6,155 judgments, with 2820 words on average, delivered by New Zealand courts from 2014 to 2021 (Wang et al., 2023)[3].

GAO and NZJD include the ground truth summary for each part. In contrast, CNN/DM only contains the ground truth for the whole document, thereby can only be used in SKGSum-W.

Baselines. We compare with the following baseline models: (1) *Traditional Summarizers*, including BART (Lewis et al., 2020)[4], MemSum (Gu et al., 2022), Seqo (Xu et al., 2022b), and GSum (Dou et al., 2021); (2) *Summarizers using document structure*, including ExtSum-LG (Xiao and Carenini, 2019) and Hiergnn (Qiu and Cohen, 2022); (3) *Summarizers using summary structure*, including IPS (Wang et al., 2023) and SSE (Wang et al., 2023); (4) *Large Language Models*, including GPT-series models (e.g., GPT-3 (Brown et al., 2020), GPT-3.5, and GPT-4 (OpenAI, 2023)) and Llama2 (Touvron et al., 2023).

Given the substantial cost of utilizing GPT, evaluations are performed on randomly selected subsets of the three datasets.

Key Phrases Extraction for Summary Points . Key phrases associated with summary points can be determined automatically using Large Language Models, such as ChatGPT. Initially, we randomly select several summaries from the same document type and use them as inputs to ChatGPT with the prompt, "Find a common and simple text structure in these examples. Based on the analysis of the provided text structure, provide ten possible high-frequency words for each part, excluding common

[3]All cases are accessible via the New Zealand Legal Information Institute website: http://www.nzlii.org/

[4]In this paper, we utilize the BART-base for efficiency considerations.

Table 3: Overall comparison on CNN/DM, NZJD, and GAO datasets. The backbone model of our methods is BART. The improvements of our methods compared to the best baseline are statistically significant (with p-value< 0.01 in paired t-tests).

Model	Structure	NZJD		GAO		CNN/DM	
		ROUGE-1/2/L	BERTScore	ROUGE-1/2/L	BERTScore	ROUGE-1/2/L	BERTScore
Llama2-7b	✗	35.12 / 12.25 / 26.46	83.07	34.42 / 15.68 / 19.38	86.02	39.95 / 15.05 / 32.69	87.26
Llama2-13b	✗	32.62 / 11.42 / 26.10	80.97	38.64 / 16.85 / 22.66	83.97	39.01 / 14.49 / 32.07	85.91
MemSum	✗	39.87 / 14.48 / 36.47	83.25	54.41 / 24.77 / 23.36	85.50	40.67 / 18.16 / 36.91	86.88
BART	✗	38.35 / 17.74 / 34.45	85.06	44.23 / 17.28 / 15.27	87.51	41.29 / 19.10 / 38.65	88.01
SeqCo	✗	32.50 / 9.87 / 20.96	82.87	37.25 / 12.73 / 18.08	84.98	41.37 / 18.61 / 28.03	87.93
GSum	✗	43.57 / 20.97 / 34.45	85.05	55.24 / **24.84** / 27.73	87.60	42.28 / 19.91 / 39.68	88.33
ExtSum-LG	✓	33.78 / 12.15 / 29.47	81.78	44.62 / 22.17 / 26.11	85.71	40.08 / 17.57 / 29.56	86.65
Hiergnn	✓	29.40 / 9.55 / 20.77	82.52	34.01 / 10.87 / 16.89	82.21	35.13 / 14.19 / 24.07	86.60
IPS	✓	39.93 / 19.20 / 37.28	85.06	55.46 / 22.87 / 27.85	87.02	N/A	N/A
SSE	✓	44.13 / 22.01 / 38.38	85.94	55.22 / 23.81 / 28.29	87.22	N/A	N/A
SKGSum *(ours)*	✓	**45.61 / 22.89** / 39.52	**86.14**	55.49 / 24.33 / **28.50**	87.27	N/A	N/A
SKGSum-W *(ours)*	✓	44.87 / 22.72 / **40.42**	86.06	**55.93** / 24.39 / 28.36	**87.76**	**44.21 / 21.47 / 41.49**	**88.48**

stop words". GPT provides an initial summary structure and each section's key phrases. Since SKGSum and SKGSum-W update the key phrase representation during training, they are robust to minor errors from ChatGPT. Experiments regarding the robustness are discussed in Section 4.3.

Evaluation Metrics . We employ ROUGE-1, ROUGE-2, and ROUGE-L, the prevalent evaluation metrics in text-generation tasks, to assess the performance of all methods. We utilize the summary-level versions of ROUGE-L, following the MemSum (Gu et al., 2022). Besides ROUGE scores, we employ BERTScore (Zhang et al., 2020) as an evaluation metric to match the generated summary and gold summary by their latent semantic similarity.

Hyper-parameter Setting. For the proposed structure-guided document encoder, the number of heads in MHP is set to 8, and all learnable parameters are initialized by a standard normal distribution. The hidden vector dimensions $d' = d$ are set to the same value as the backbone encoders or decoders.

4.2 Overall Comparisons

Table 3 presents a comparison between previous representative summarizers, large language models, and our proposed methods. Table 4 demonstrates the effectiveness of SKGSum and SKGSum-W when applied to existing encoder-decoder summarizers. From the results, we can conclude:

- **Compared to the conventional methods**, methods that consider structure information significantly outperform conventional methods. This is because the conventional methods may lose information regarding some summary points as illustrated in Table 1.

Table 4: Results on New Zealand Judgment Dataset. The p-value of the t-test between SKGSum and the best baseline model on overall scores is less than 0.05.

Version	Chunks	MemSum		BART	
		ROUGE-1/2/L	BS	ROUGE-1/2/L	BS
Vanilla	Overall	39.87/14.48/36.47	82.35	38.35/17.74/34.45	85.06
IPS	Overall	40.70/15.91/37.56	83.13	39.93/19.20/37.28	85.07
	-Decision	31.98/14.12/28.50	84.74	63.27/48.85/61.99	91.61
	-Fact	33.24/13.17/29.99	83.74	32.53/13.44/28.01	84.95
	-Reason	31.44/11.80/28.41	83.90	25.37/9.98/23.31	84.02
SSE	Overall	41.63/16.58/39.07	82.35	44.13/22.01/38.38	85.94
	-Decision	32.88/14.51/29.38	85.13	66.31/51.26/64.47	92.05
	-Fact	35.24/14.62/31.93	83.16	36.47/16.31/27.10	85.63
	-Reason	32.43/12.52/29.47	83.43	29.28/11.40/23.47	84.83
SKGSum	Overall	**42.18/17.00/39.55**	**83.18**	**45.61/22.89**/39.52	**86.14**
	-Decision	33.12/14.89/29.77	85.12	65.71/50.74/63.65	91.89
	-Fact	35.69/14.86/32.35	83.74	37.66/17.12/28.05	85.72
	-Reason	33.02/12.96/29.95	83.90	31.34/12.86/25.08	85.00
SKGSum-W	Overall	41.47/16.36/38.78	82.61	44.87/22.72/**40.42**	86.06

- **Compared to methods that leverage summary structure**, SKGSum and its variant outperform the state-of-the-art method SSE (Wang et al., 2023) in generating the whole summary, as well as each individual summary point. Notably, SKGSum shows significant advancements in lengthier sections, like *Fact* and *Reason* on the NZJD dataset. This superior performance can be attributed to SKGSum's ability to constrain the content of each summary section, informed by the entirety of the guidance information and the overall summary structure.

- **Comparing SKGSum and SKGSum-W** in Tables 3 and 4, SKGSum generally scores slightly higher than SKGSum-W. However, it's crucial to remember that SKGSum-W does not require labels for each part of the gold summary.

4.3 Sensitivity to the Extracted Key Phrases

Since key phrases for summary points extracted by ChatGPT may vary, it is crucial to assess the effect of the summary point representation initialized by the extracted key phrases. To this end, we compare the summary point representations obtained

Table 5: Effect of Summary Point Representation on the NZJD with BART-SKGSum (ROUGE-1/ROUGE-2/ROUGE-L).

	Proposed Key Phrases for Summary Points	Random
Overall	45.61 / 22.89 / 39.52	45.05 / 22.60 / 39.23
Decision	65.71 / 50.74 / 63.65	65.86 / 51.36 / 64.05
Fact	37.66 / 17.12 / 28.05	37.45 / 17.04 / 27.94
Reason	31.34 / 12.86 / 25.08	30.18 / 12.28 / 24.60

from the extracted key phrases (as detailed in Section 4.1) with random initialization. We test with **SKGSum** on the New Zealand Judgment dataset and report the results in Table 5. Notably, the GPT-based key phrase extraction process enhances the ROUGE scores for most of the summary sections, albeit the differences are subtle. This can be attributed to the trainability of the summary point representation. However, the GPT-based keyphrase extraction method may offer a better initialization, leading to better results.

Additionally, we evaluate GPT's capability to capture the structure of summaries, with the details of this test outlined in Appendix B. After comparison between GPT observed summary structures with reference structure labels, we obtain 83% F1 scores and 0.26 Hamming Loss in CNN/DM datasets; 80% F1 scores and 0.31 Hamming Loss for Government reports test; and 79% F1 scores and 0.32 Hamming Loss for New Zealand Judgment comparisons. The high F1 and low Hamming Loss represent the GPT's ability to mine the summary structure for input summaries. However, it is noteworthy that while GPT is adept at mining structures, it cannot generate high-quality summaries because GPT does not leverage the summary structure.

4.4 Case Study

Table 1 presents summaries generated by SKGSum using BART (Lewis et al., 2020) as the backbone.

SKGSum notably enhances the abstractive summarizers. The resultant summary encompasses all summary points with essential information. Compared to the knowledge-guided summarizer, GSum, SKGSum produces more structured and complete summaries, indicating the efficacy of the knowledge utilized in our method for generating news summaries. When compared to large language models such as GPT-4 and Llama2, SKGSum yields summaries that are well-structured and complete. The results necessitate a specialized design for structured summarization. Similar findings can be observed in another case study on judgment data (Appendix C).

4.5 Comparison to Large Language Models

We compare our model with two most recent versions of GPT-3.5 models[5], Gpt-3.5-turbo and text-davinci-003, and GPT-4[6]. Among them, GPT-3.5-turbo is tailored for handling dialogue, while text-davinci-003 is designed for general NLP tasks.

In this experiment, we randomly selected 50 cases from the test sets of each dataset. The results, presented in Table 6, show the superiority of SKGSum over the GPT-series models owing to its specialized design for structured summarization. An in-depth discussion of this comparison can be found in Appendix D.

5 Related Work

Extractive and Abstractive Summarizers. Existing summarizers can be roughly divided into two types: extractive and abstractive summarizers.

Extractive summarization classifies the sentences in the source document and determines whether a sentence should be included in the summary. NeuralSum (Cheng and Lapata, 2016) is one of the earliest methods using Encoder-Decoder structure. Recent research like ExtSum-LG (Xiao and Carenini, 2019) uses the RNN-based encoder for multi-level representation of sentences by considering document and sentence structure. Then, BERT encoder (Liu and Lapata, 2019b; Sotudeh and Goharian, 2022; Liu and Lapata, 2019b) and Transformer encoder (Zhang et al., 2019; Ruan et al., 2022; Zhang et al., 2019) are introduced to replace RNN-related encoders. Besides, MemSum (Gu et al., 2022) introduces reinforcement learning to reduce redundancy in generating summaries by considering selection histories.

Abstractive methods generate summaries word by word. BERT (Devlin et al., 2019) and BART (Lewis et al., 2020) are representative models in this type. Based on these models, recent studies introduce new knowledge to improve the summarization performance, such as Document Segmentation (Moro and Ragazzi, 2022), Role Labeling Detection (Elaraby and Litman, 2022),

[5] https://platform.openai.com/docs/models/gpt-3-5

[6] https://platform.openai.com/docs/models/gpt-4

Table 6: Comparison to GPT. BS denotes the BERTScore. The backbone model of our proposed methods is BART.

	Model	CNN/DM		NZJD		GAO	
		ROUGE-1/2/L	BS	ROUGE-1/2/L	BS	ROUGE-1/2/L	BS
Normal	text-davinci-003	38.48 / 15.42 / 32.70	87.36	31.12 / 9.37 / 18.81	82.63	22.72 / 10.50 / 14.12	85.92
	GPT-3.5-turbo	37.52 / 15.42 / 31.03	87.32	34.98 / 11.22 / 21.59	83.34	25.44 / 11.93 / 15.19	86.42
	GPT-4	38.32 / 15.39 / 31.65	87.58	28.65 / 8.39 / 16.85	82.57	27.71 / 10.61 / 15.42	86.11
Structure	text-davinci-003	N/A	N/A	37.08 / 11.80 / 22.68	82.37	46.06 / 19.42 / 26.43	86.83
	GPT-3.5-turbo	N/A	N/A	35.26 / 11.72 / 21.92	82.27	47.41 / 19.91 / 25.87	86.89
	GPT-4	N/A	N/A	31.74 / 8.54 / 24.26	81.67	39.13 / 14.18 / 21.31	85.92
	SKGSum	N/A	N/A	**43.42 / 20.78 / 31.20**	85.44	53.41 / 22.75 / 27.31	87.14
	SKGSum-W	**42.94 / 21.74 / 40.99**	**88.21**	41.82 / 19.27 / 28.92	**85.65**	**54.91 / 23.35 / 27.67**	**87.71**

Entity Aggregation (González et al., 2022), Key Phrases Detection (Liu et al., 2021), and Guidance Signals (Dou et al., 2021).

Knowledge-Guided Summarization. Recent research shows that introducing guiding information into summarizers can improve summarization quality. We refer to this approach as Knowledge-guided summarization, which combines aspects of both aspect-based and query-based methods. Because the guiding information can be any type, such as keywords (Li et al., 2018a), closest summary (Cao et al., 2018), key sentences (Wang et al., 2022) from the source document, salience allocation expectation (Wang et al., 2022) or relational triples (Jin et al., 2020). GSum (Dou et al., 2021) proposed a general and scalable guided summarization framework based on previous work, which can accept various signals like highlighted sentences to generate summaries, keywords, and relational triples as Guiding Knowledge to guide summary generation.

Structured Summarization. Two types of structure have been utilized in the literature: document structure and summary structure. Most existing works explored common document structures, such as source syntactic structures (Song et al., 2018) and document organizations (Li et al., 2018b). Recent studies have attempted to leverage summary structure, including: (1) methods that align the source document's structure with that of the summary (Frermann and Klementiev, 2019; Balachandran et al., 2021; Cao and Wang, 2022); and (2) methods that leverage human predefined summary structure labels (Wang et al., 2023). However, these methods either rely on strong assumptions about the source document structure or heavily depend on human-annotated summary structures, limiting their applicability to diverse real-world documents.

6 Conclusion

In this paper, we analyze summaries of lengthy documents and observe that their summaries consistently adhere to a distinct structure. To harness this structural information, we introduce two methods, namely SKGSum and SKGSum-W. Our methods align their generated summaries closely with automatically extracted summary points. Experiments on three real-world open-source datasets have demonstrated the superiority of our method over state-of-the-art methods in summarizing news articles, legal judgments, and government reports.

Limitations

SKGSum achieves high performance by utilizing summary structure guiding knowledge, but this introduces additional work, such as using GPT to analyze the structure. Furthermore, although SKGSum-W shows promising capability in generating high-quality summaries without specific part labels from the gold summary, its performance in some scenarios still falls short when compared to SKGSum. These two aspects present research opportunities that we aim to explore in future work.

Acknowledgment

This research is supported by the Marsden Fund Council from Government funding (MFP-UOA2123), administered by the Royal Society of New Zealand.

References

Vidhisha Balachandran, Artidoro Pagnoni, Jay Yoon Lee, Dheeraj Rajagopal, Jaime Carbonell, and Yulia Tsvetkov. 2021. StructSum: Summarization via structured representations. In *EACL*, pages 2575–2585.

Katrina Banks-Smith. 2019. More than just precedent: Perspectives on judgment writing. In *University of*

Notre Dame Australia Law Review, volume 21, pages 1–17.

Gillian Brown and George Yule. 1983. *Discourse analysis*. Cambridge university press.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*.

Peter Butt. 2013. Judgment writing: an antipodean response. In *Law quarterly review*, volume 129, pages 7–10. Sweet Maxwell Ltd. (UK).

Shuyang Cao and Lu Wang. 2022. HIBRIDS: attention with hierarchical biases for structure-aware long document summarization. In *ACL*, pages 786–807.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *ACL*, pages 152–161.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *arXiv*.

Prafulla Kumar Choubey, Aaron Lee, Ruihong Huang, and Lu Wang. 2020. Discourse as a function of event: Profiling discourse structure in news articles around the main event. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *NAACL*, pages 4830–4842.

Mohamed Elaraby and Diane Litman. 2022. ArgLegalSumm: Improving abstractive summarization of legal documents with argument mining. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6187–6194.

Lea Frermann and Alexandre Klementiev. 2019. Inducing document structure for aspect-based summarization. In *ACL*, pages 6263–6273.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. Structured summarization of academic publications. In *Machine Learning and Knowledge Discovery in Databases*, pages 636–645.

José Ángel González, Annie Louis, and Jackie Chi Kit Cheung. 2022. Source-summary entity aggregation in abstractive summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6019–6034.

Nianlong Gu, Elliott Ash, and Richard Hahnloser. 2022. MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes. In *ACL*, pages 6507–6522.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*.

Margaret Hill. 1991. Writing summaries promotes thinking and learning across the curriculum: But why are they so difficult to write? In *Journal of reading*, volume 34, pages 536–539.

Luyang Huang, Shuyang Cao, Nikolaus Nova Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *NAACL*, pages 1419–1436.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Semsum: Semantic dependency guided neural abstractive summarization. In *AAAI*, pages 8026–8033.

David M Keepnews. 2016. Developing a policy brief. *Policy, Politics, & Nursing Practice*, 17(2):61–65.

Dennis Kurzon. 1985. How lawyers tell their tales: Narrative aspects of lawyer's brief. *Poetics*, 14(6):467–481.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880.

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. 2018a. Guiding generation for abstractive text summarization based on key information guide network. In *NAACL*, pages 55–60.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018b. Improving neural abstractive document summarization with structural regularization. In *EMNLP*, pages 4078–4087.

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

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *ACL*, pages 5070–5081.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *EMNLP*, pages 3730–3740.

Yang Liu, Chenguang Zhu, and Michael Zeng. 2022. End-to-end segmentation-based news summarization. In *ACL Findings*, pages 544–554.

M Makdisi and J Makdisi. 2009. How to write a case brief for law school: Excerpt reproduced from introduction to the study of law: Cases and materials. *LexisNexis*.

James R Martin. 1992. *English text: System and structure*. John Benjamins Publishing.

Gianluca Moro and Luca Ragazzi. 2022. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. In *AAAI*, volume 36, pages 11085–11093.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

OpenAI. 2023. GPT-4 technical report. *arxiv*, abs/2303.08774.

Jason Phang, Yao Zhao, and Peter Liu. 2023. Investigating efficiently extending transformers for long input summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3946–3961, Singapore.

Yifu Qiu and Shay B. Cohen. 2022. Abstractive summarization guided by latent hierarchical document structure. In *EMNLP*, pages 5303–5317.

Mathieu Ravaut, Shafiq Joty, and Nancy Chen. 2022. SummaReranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization. In *ACL*, pages 4504–4524.

Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. HiStruct+: Improving extractive text summarization with hierarchical structure information. In *ACL Findings*, pages 1292–1308.

Kaiqiang Song, Lin Zhao, and Fei Liu. 2018. Structure-infused copy mechanisms for abstractive summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1717–1729.

Sajad Sotudeh and Nazli Goharian. 2022. TSTR: Too short to represent, summarize with details! intro-guided extended summary generation. In *ACL*, pages 325–335.

John M Swales and John Swales. 1990. *Genre analysis: English in academic and research settings*. Cambridge university press.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arxiv*, abs/2307.09288.

Fei Wang, Kaiqiang Song, Hongming Zhang, Lifeng Jin, Sangwoo Cho, Wenlin Yao, Xiaoyang Wang, Muhao Chen, and Dong Yu. 2022. Salience allocation as guidance for abstractive summarization. In *arXiv*.

Qiqi Wang, Ruofan Wang, Kaiqi Zhao, Robert Amor, Benjamin Liu, Xianda Zheng, Zeyu Zhang, and Zijian Huang. 2023. Towards legal judgment summarization: A structure-enhanced approach. In *ECAI 2023 - 26th European Conference on Artificial Intelligenc*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 2491–2498.

Penny Whiting, Mariska Leeflang, Isabel de Salis, Reem A Mustafa, Nancy Santesso, Gowri Gopalakrishna, Geraldine Cooney, Emily Jesper, Joanne Thomas, and Clare Davenport. 2018. Guidance was developed on how to write a plain language summary for diagnostic test accuracy reviews. In *Journal of Clinical Epidemiology*, volume 103, pages 112–119.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *EMNLP*, pages 3011–3021.

Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. 2022a. Sequence level contrastive learning for text summarization. In *AAAI*, volume 36, pages 11556–11565.

Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. 2022b. Sequence level contrastive learning for text summarization. In *Thirty-Sixth AAAI Conference on Artificial Intelligence*, pages 11556–11565.

Yu-Fen Yang. 2015. Automatic scaffolding and measurement of concept mapping for efl students to write

summaries. In *Journal of Educational Technology & Society*, volume 18, pages 273–286.

Zonghai Yao, Benjamin Schloss, and Sai Selvaraj. 2023. Improving summarization with human edits. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2604–2620, Singapore.

Li Yuan ke and Michael Hoey. 2014. Strategies of writing summaries for hard news texts: A text analysis approach. In *Discourse studies*, volume 16, pages 89–105.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR*.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *ACL*, pages 5059–5069.

A Loss Functions

Extractive summarizers select sentences from the source document and utilize classification loss:

$$\mathcal{L} = -\sum_{(D_i,S_i)\in\mathcal{D}_{train}} \sum_{k=1}^{p}\sum_{j=1}^{m} \lambda \log p(s_{ij}^k|g_{ij}^k) + \left(1-s_{ij}^k\right)\log p\left(s_{ij}^k|g_{ij}^k\right), \quad (6)$$

where g_{ij}^k is the decoder output and probability for selecting the j-th sentence of document D_i for the k-th part, and s_{ij}^k is the binary ground-truth label of whether the sentence should be in the gold summary of D_i. If the j-th sentence is selected then $s_{ij}^k = 1$, otherwise, $s_{ij}^k = 0$. The parameter λ is a weight to balance the importance of selected and unselected sentences in the loss function. Choosing a larger value of λ can mitigate the issue of imbalanced classes by boosting the importance of selected sentences. When $\lambda = 1$, the selected and unselected sentences are of the same importance (Liu and Lapata, 2019a).

As for **abstractive summarizers**, the loss is

$$\mathcal{L} = -\sum_{(D_i,S_i)\in\mathcal{D}_{train}} \sum_{k=1}^{p}\sum_{j=1}^{m} s_{ij}^k \log p(s_{ij}^k|g_{ij}^k), \quad (7)$$

where s_{ij}^k and g_{ij}^k are the j-th word token in the gold summary and generated summary of the k-th summary part for document D_i.

B Evaluation of GPT Capturing Summary Structure

GPT models are utilized to capture the structural essence of summaries, a task traditionally managed through human design, in an effort to maintain robustness. Therefore, we introduce additional tests to evaluate GPT's capabilities in this regard.

Evaluation Formulation. Given summary S, and the reference summary structure comprising several parts, $s_1, s_2, \cdot, s_p$, and ChatGPT-4 concluded summary structure, $\hat{s_1}, \hat{s_2}, \cdot, \hat{s_k}$. The challenge lies in determining the alignment between GPT-4's generated summary structure, $\hat{s_j}$, and aligning the reference summary structure s_i.

This issue can be simplified into a multi-label classification problem. Assuming we have P possible summary points, where $P = max(p, k)$. The objective is to assess the accuracy with ChatGPT-4 giving the same part labels.

Evaluation Metrics. To evaluate the performance of GPT-4 in accurately classifying summary points, we employ two well-known multi-label classification metrics: the F1 Score and Hamming Loss. The F1 Score ranges from 0 to 1, where 1 indicates perfect result, and 0 indicates the worst performance. In contrast, a lower Hamming Loss signifies better model performance, with 0 being the ideal score.

Dataset. We continue utilizing the previously employed datasets for evaluating the generated summaries' quality, including CNN/DM, GAO, and NZJD. The GAO and NZJD datasets inherently contain structured summaries. For the CNN/DM dataset, we engaged an expert to delineate the summary points conclusively. Table 2 presents the summary structures for all three datasets.

Subsequently, we randomly constructed 50 test samples for each dataset, with each sample comprising 20 summaries. To identify the summary structure within these samples, we employed ChatGPT-4 using the same prompt as described in Section 4.1: "Find a common and simple text structure in these examples".

Results. After comparison between GPT observed summary structures with reference structure labels, we obtain 83% F1 scores and 0.26 Hamming Loss in CNN/DM datasets; 80% F1 scores and 0.31 Hamming Loss for Government reports

Table 7: Example New Zealand Judgment Summaries. The extractive, abstraction, and knowledge-guided summaries are obtained by ExtSum-LG, BART, and GSum, respectively. The Magenta, Cyan, and Brown colors represent the *Decision*, *Facts*, and *Reasons* components, respectively.

Human: Unsuccessful application by t for leave to appeal court of appeal decision. t be convict on eight representative count of historic sexual offending against his daughter and sentence to 10 years imprisonment. t unsuccessfully appeal his conviction and sentence but be unsuccessful. t now wish to appeal his conviction on ground (a) wrongfully pressure into accompany the police to the police station and that this misconduct render his trial unfair and result in a miscarriage of justice and (b) trial judge should have grant the jury's request to see the transcript of the complainant's video statement and her write statement. hold, case do not raise any issue of general principle. Also, nothing to indicate that there be any risk of a substantial miscarriage of justice.
ExtSum-LG: the trial judge rejected the jury's request following a discussion with counsel, on the ground that the statements were not in evidence and reminded the jury that they were to determine the case on the basis of the evidence adduced in court. the applicant submits that he was wrongfully pressured into accompanying the police to the police station and that this misconduct rendered his trial unfair and resulted in a miscarriage of justice. Moreover, we see nothing to indicate that there is any risk of a substantial miscarriage of justice. In the circumstances of this case, that is plainly correct. The applicant's leave submissions focus first on alleged misconduct by the police when he was taken to the police station, before he was interviewed and formally arrested.
BART: ca dismiss d's appeal against conviction for sexual violation of a child. d seek leave to appeal on basis the sentence be manifestly excessive. hold, propose appeal do not meet criterion for leave
GSum: successful application by t for leave to appeal court of appeal ('ca ') decision. ca dismiss t's appeal against conviction for sexual violation by unlawful sexual connection and indecent act on a young person. for appeal on ground ca do not give rise to miscarriage of justice. with no appearance of miscarriage of miscarriage arise in accordance with t' name, address, occupation or identify particular of justice arose from the complainant in the complainant's counsel to the complainant and the trial judge.
GPT-4: The applicant, convicted on eight counts of historic sexual abuse against his daughter, sought leave to appeal his convictions, claiming police misconduct and that the trial Judge did not allow the jury to see the complainant's video statement. The Court dismissed these claims, stating any alleged police misconduct could not have affected the trial's fairness, and it was correct not to admit the video statement as evidence. Consequently, the Court concluded there was no risk of serious injustice and dismissed the application for leave to appeal.
BART-SKGSum: unsuccessful application by t for leave to appeal court of appeal ('ca') decision. ca dismiss t's appeal against conviction for historic sexual offending against his daughter. t seek leave to appeal on ground he be wrongfully pressure into accompany the police to the police station, which render his trial unfair and result in a miscarriage of justice. hold, case do not raise any issue of general principle. no risk of miscarriage of justice.
ExtSum-LG-SKGSum: The application for leave to appeal is dismissed. The applicant submits that he was wrongfully pressured into accompanying the police to the police station and that this misconduct rendered his trial unfair and resulted in a miscarriage of justice. He now seeks leave to appeal to this Court against his convictions. He appealed to the Court of Appeal against his convictions and his sentence but was unsuccessful. He was sentenced to imprisonment for 10 years. The applicant's leave submissions focus first on alleged misconduct by the police when he was taken to the police station, before he was interviewed and formally arrested. Moreover, we see nothing to indicate that there is any risk of a substantial miscarriage of justice. This case does not raise any issue of general principle. This is consistent with this Court's recent decision in Guy v R.

test; and 79% F1 scores and 0.32 Hamming Loss for New Zealand Judgment comparisons.

Result Analysis. The results show that ChatGPT-4 can achieve approximately 80% F1 scores and a 0.3 Hamming Loss across all three datasets. This indicates that ChatGPT-4 has the capability to capture the structure of summaries. The majority of incorrect cases are caused by ChatGPT-4 providing excessive details. For example, ChatGPT-4 consistently splits "News Details," a summary point from CNN/DM, into two parts: an introduction and a description; it divides the "Fact" of NZJD into three parts, including background, legal claims, and previous decisions. Although these divisions are not incorrect, they are not suitable for all cases. For instance, some news summaries are too brief to require a two-part description of the news story.

Further, we want to highlight that while GPT is adept at mining structures, this does not necessarily equate to its ability to generate high-quality summaries. A key factor in this distinction is the length of the input texts used for structure mining, which is significantly smaller than the length of the source documents.

C Case Study Analysis

We conducted another case study on the New Zealand Judgment[7].

Table 7 shows the generated summaries by existing state-of-the-art (SOTA) summarizers, large language model, GPT-4, and our proposed method.

In terms of legal judgments, compared with human-written summaries, existing state-of-the-art (SOTA) summarizers cannot properly cover three essential categories consisting of "Decision", "Facts", and "Reasons". Specifically, extractive and abstractive summarizers lose information on "Decisions" and are less focused on "Reasons", while the guiding knowledge method (Dou et al., 2021) loses details of "Reasons" and makes mistakes in

[7]The case is shown on the web: http://www.nzlii.org/cgi-bin/sinodisp/nz/cases/NZSC/2015/9.html

Table 8: GPT-series Comparison. davinci represents the text-davinci-003; and GPT-3.5 is GPT-3.5-turbo. The backbone model of our proposed methods in these comparisons is BART. BS denotes the BERTScore.

Type	Model	Chunks	CNN/DM ROUGE-1/2/L	CNN/DM BS	NZJD ROUGE-1/2/L	NZJD BS	GAO ROUGE-1/2/L	GAO BS
Normal	davinci	Overall	38.48 / 15.42 / 32.70	87.36	31.12 / 9.37 / 18.81	82.63	22.72 / 10.50 / 14.12	85.92
	GPT-3.5	Overall	37.52 / 15.42 / 31.03	87.32	34.98 / 11.22 / 21.59	83.34	25.44 / 11.93 / 15.19	86.42
	GPT-4	Overall	38.32 / 15.39 / 31.65	87.58	28.65 / 8.39 / 16.85	82.57	27.71 / 10.61 / 15.42	86.11
Structure	davinci	Overall	N/A	N/A	37.08 / 11.80 / 22.68	82.37	46.06 / 19.42 / 26.43	86.83
		-Part 1	N/A	N/A	23.16 / 8.37 / 18.86	84.24	50.08 / 25.92 / 34.46	88.83
		-Part 2	N/A	N/A	32.65 / 9.79 / 19.90	83.10	28.50 / 9.55 / 16.14	85.37
		-Part 3	N/A	N/A	26.89 / 5.35 / 15.29	82.81	30.99 / 6.61 / 20.23	86.56
	GPT-3.5	Overall	N/A	N/A	35.26 / 11.72 / 21.92	82.27	47.41 / 19.91 / 25.87	86.89
		-Part 1	N/A	N/A	13.58 / 4.80 / 10.34	83.19	49.87 / 23.05 / 30.60	88.32
		-Part 2	N/A	N/A	34.24 / 11.02 / 20.93	83.55	29.69 / 10.21 / 16.31	85.35
		-Part 3	N/A	N/A	25.16 / 5.45 / 15.33	82.47	32.98 / 7.67 / 21.24	87.09
	GPT-4	Overall	N/A	N/A	31.53 / 8.37 / 18.66	81.67	39.13 / 14.18 / 21.31	85.92
		-Part 1	N/A	N/A	15.18 / 4.79 / 12.37	83.55	45.69 / 16.75 / 27.18	87.85
		-Part 2	N/A	N/A	30.03 / 8.14 / 18.02	82.97	26.33 / 7.52 / 14.13	84.39
		-Part 3	N/A	N/A	15.31 / 1.62 / 10.05	81.63	19.02 / 2.48 / 12.38	85.19
	SKGSum	Overall	N/A	N/A	**43.42 / 20.78 / 31.20**	85.44	53.41 / 22.75 / 27.31	87.14
		-Part 1	N/A	N/A	65.57 / 51.88 / 63.484	92.07	57.43 / 32.91 / 41.01	90.10
		-Part 2	N/A	N/A	37.33 / 16.37 / 27.09	85.31	43.67 / 13.82 / 20.09	85.57
		-Part 3	N/A	N/A	30.10 / 10.15 / 21.57	84.50	32.32 / 10.10 / 25.29	88.24
Structure w/o Label	SKGSum-W	Overall	**42.94 / 21.74 / 40.99**	**88.21**	42.76 / 20.20 / 30.82	**85.65**	**54.91 / 23.35 / 27.67**	**87.71**

"Decisions". A similar finding can be found in the GPT-4 generated summary which misses the "Reasons" part. Because these models, especially guiding knowledge summarizers, focus solely on the most frequent information. Furthermore, current extractive summarizers tend to interleave different types of information, which reduces the generated summaries' readability. Besides, summary sections can be relevant, and a sentence can be relevant to multiple summary parts. The current models do not capture the relevance across summary sections.

The bottom of Table 7 displays summaries generated by employing two backbone summarizers, ExtSum-LG (Gu et al., 2022) and BART (Lewis et al., 2020), integrated with **SKGSum**.

In comparison to results from prior models, the **SKGSum** adaptation of ExtSum-LG incorporates *Reasons*, rendering the generated summary more comprehensive. Additionally, summaries generated by the standalone ExtSum-LG are disordered and perplexing, making it challenging for readers to comprehend the central points of the summary. The integration of the **SKGSum** methodology significantly refines the abstractive summarizer, yielding a summary that encompasses all anticipated summary points and, notably, compared to summaries generated by BART, conveys more crucial information in the **SKGSum** version. These enhancements underscore the efficacy of the **SKGSum** methodology in refining existing summarizers.

Moreover, when compared to the knowledge-guided summarizer, GSum, **SKGSum** produces summaries that are organized, comprehensive, and accurate, indicating that the knowledge employed in our method is efficacious for judgment summaries.

Lastly, when compared with the large language model-based summarizer, GPT-4, **SKGSum** yields summaries that are well-structured, complete, and balanced. It shows that the specialized design of structured summaries is essential, as the summary quality is better than that of the generalized language model for generating judgment summaries.

D GPT-series Comparison Detail

We conduct comparisons with GPT-series models.

We select two versions of GPT-3.5 models [8], text-davinci-003 and GPT-3.5-turbo, and GPT-4. Gpt-3.5-turbo is able to handle communication, and text-davinci-003 is focused on general NLP tasks.

We randomly chose 50 test cases per dataset. This means we will ask the GPT to generate summaries for 150 documents belonging to three different types of source documents (News, Judgments, and government reports).

Then, we set up two tests. First, we utilize GPT to generate the summary directly, in a way similar to what most previous summarizers did: without summary structure. We follow the GPT Official Document, which provides instructions and examples to utilize *Tl;dr* as prompts. Second, we utilize the summary point designed in Section 4.1 as prompts to generate structured summaries.

Table 8 shows the results. There are a few findings:

[8] https://platform.openai.com/docs/models/gpt-3-5

- The proposed idea of structure summarization is effective. Our structure summarization approach improves the performance of GPT models by generating summaries part by part. When utilizing the designed summary points, we see a notable enhancement in the overall results. This improvement also can be found in Tables 3, and 4.

- Another key observation is that our proposed method SKGSum-W, which doesn't necessitate part summaries, still outperforms both the GPT and its structured counterpart.

Furthermore, the cost is another factor in applying models in the real world. In total, the comparison costs US$ 76.45 to generate summaries for the 200 cases. If we had tested all the test sets, the total costs would have been more than US$ 700. This suggests that using GPT to do the summarization task can be a high-cost but low-performance choice.

In summary, this comparison shows that both proposed structure summarization methods are effective and essential, whether we use large language models to sum up long documents or not.