Journal of King Saud University Computer and Information Sciences (2025) 37:40
https://doi.org/10.1007/s44443-025-00041-2

ORIGINAL PAPER

CoTHSSum: Structured long-document summarization via chain-of-thought reasoning and hierarchical segmentation

Xiaoyong Chen[1,2] · Zhiqiang Chen[2] · Shi Cheng[1]

Received: 1 April 2025 / Accepted: 21 April 2025 / Published online: 20 May 2025

Abstract
Long-document summarization remains a challenging task for large language models (LLMs), which often suffer from input length constraints, semantic incoherence, and factual hallucinations when processing extensive and complex texts. In this paper, we propose a novel summarization framework that integrates hierarchical input segmentation with Chain-of-Thought (CoT) prompting to guide LLMs through structured, interpretable reasoning. Our method decomposes long documents into semantically coherent segments, applies CoT-based prompting for intermediate summary reasoning, and employs structure-guided decoding to compose high-quality final summaries. We evaluate our approach across five diverse datasets, including scientific, biomedical, governmental, literary, and legal domains, using strong LLM backbones such as Qwen, LLaMA, and Phi. Experimental results demonstrate that our method consistently outperforms state-of-the-art baselines across ROUGE, BLEU, BERTScore, and factual consistency metrics. Ablation and human evaluation further confirm the complementary benefits of CoT reasoning and hierarchical structure, offering a reliable and scalable solution for summarizing complex long-form content.

Keywords Natural language generation · Natural language processing · Chain-of-thought · Hierarchical structuring · Factual consistency

1 Introduction

The rapid growth of online text in news, scholarly articles, and social media has made automatic summarization a crucial technology for information management (Zhang et al 2024; Altmami and Menai 2022; Kerui et al 2020). Summarization is generally defined as the task of distilling the most important information from a source document to produce a concise and coherent version for users (Wibawa et al 2024; Bao et al 2025). While summarization has long been studied, summarizing lengthy documents remains especially challenging. As text length grows, models not only risk semantic drift but also struggle to preserve crucial content. Early works thus explored hierarchical architectures to maintain coherence and capture broader context (Deroy et al 2024; Tsirmpas et al 2024; Yu et al 2025). For instance, Ou et al. employed a hierarchical encoder-decoder to better capture long-range context in scientific papers, highlighting the need for structure-aware models in summarizing lengthy texts (Ou and Lapata 2025). These challenges underscore that summarizing long documents is not a trivial extension of short-text summarization-it requires handling information at multiple levels of granularity and preserving the document's logical flow.

Recent advances in Large Language Models (LLMs), such as GPT-3 (Floridi and Chiriatti 2020) and GPT-4 (Achiam et al 2023), have substantially elevated summarization performance. Leveraging massive pre-training, these models excel at zero- or few-shot summarization, prompting some researchers (Pu et al 2023) to suggest that "summarization is (almost) dead.? However, as input length increases, LLM-based summarization still faces limitations in factual consistency and coverage. Implying that LLMs have essentially solved the core summarization task in many settings. Despite these advances, the biggest challenges lie in simultaneously maintaining factual consistency and comprehensive coverage. Current techniques, especially when dealing with

✉ Shi Cheng
chengshi@email.cn

1 School of Artificial Intelligence and Computer Science, Nantong University, Nantong, Jiangsu 226000, China

2 Information Science Department, Nantong University Xinglin College, Nantong, Jiangsu 226000, China

extensive inputs, frequently fail to ensure that all critical segments are accurately included while avoiding hallucinations.

Existing LLM-based summarizers often produce hallucinations, where generated content is factually incorrect or ungrounded (Yang et al 2023; Zhang et al 2023), and may omit critical content when dealing with very long inputs (Khan et al 2023). Even hierarchical or chunk-based approaches can yield disjointed results without careful coordination (Edge et al 2024; Huang et al 2021). LLMs operating under fixed context windows may overlook or omit key points from the tail of a document, leading to incomplete summaries (Khan et al 2023). Researchers have observed that naive attempts at long-text summarization can truncate important content, prompting the exploration of methods to extend a model's memory or to split the task into smaller parts (Chang et al 2023; Edge et al 2024; Huang et al 2021). For example, one approach is to recursively summarize portions of a text (such as individual chapters of a book) and then summarize those summaries, to incrementally cover the full document. This kind of hierarchical process can mitigate context length limitations, but without careful coordination it may still yield a disjoint final summary. In general, long and complex inputs can cause a summary to lose logical structure or cohesion, as the model may struggle to organize information from across the document into a well-structured narrative. Without an explicit reasoning or planning mechanism, the summary may read as a set of loosely connected points rather than a coherent whole.

In this paper, we propose a new method for long-text summarization that addresses these issues by combining chain-of-thought (CoT) guided reasoning with a hierarchical input/output structure. Chain-of-thought prompting involves guiding the LLM to generate intermediate reasoning steps or sub-summaries, rather than directly jumping to the final summary (Zhang et al 2024; Wang et al 2023; Liu et al 2024; Choi et al 2025).Prior work has shown that CoT reasoning can elicit more logical and internally consistent responses from LLMs on complex tasks. In the context of summarization, a step-by-step reasoning approach encourages the model to integrate fine-grained details and maintain factual consistency at each step. Notably, a recent study (Zhu et al 2025) introduced a "summary chain-of-thought" prompting strategy, guiding GPT-series models to generate summaries iteratively; this was found to reduce hallucinations and improve the inclusion of important details in the final output. Building on this insight, our method uses CoT to decompose the summarization process: the LLM first produces structured intermediate representations (such as outlines, key point lists, or section summaries) which are then used to construct the final summary. This is coupled with a hierarchical summarization framework, wherein the long input document is broken into smaller coherent segments. The model summarizes each segment (potentially with its own chain-of-thought), and these segment summaries are subsequently aggregated and refined. By organizing the input and generation in a hierarchy, the model can focus on one part of the document at a time, greatly reducing context fragmentation and ensuring that no important segment is overlooked. The higher-level reasoning (guided by CoT) then ties together the segment summaries, which preserves global coherence and logical flow in the ultimate summary. Through this combined CoT and hierarchical approach, our method aims to achieve more faithful and well-structured summaries for long texts than existing one-shot LLM summarization. In what follows, we detail the implementation of our approach and evaluate it on a variety of long-document summarization benchmarks. We show that it substantially alleviates common LLM limitations-reducing factual hallucinations, improving coverage of key information, and producing more organized summaries-all without requiring any task-specific fine-tuning of the base language model.

The core motivation of our research is to address two pressing limitations in LLM-based summarization of long documents: (1) mitigating hallucinations through structured, step-by-step reasoning, and (2) ensuring no critical content is overlooked when segmenting extensive inputs. By integrating Chain-of-Thought (CoT) prompting with hierarchical input segmentation, we encourage the LLM to gradually refine its summary across manageable chunks, thereby preserving factual consistency and logical flow. This joint strategy fills a gap in current methods, which often overlook nuanced details or fail to maintain overall coherence when summarizing large-scale texts. The contributions of this paper are threefold:

- We propose a novel summarization framework that integrates chain-of-thought reasoning with a hierarchical summarization strategy. To our knowledge, this is the first approach to combine stepwise logical reasoning and multi-level input segmentation for long-text summarization by LLMs.
- We demonstrate that our CoT-guided hierarchical method outperforms standard prompting techniques on long-document summarization tasks. It yields summaries with higher factual consistency, better preservation of important content, and improved structural coherence, as evidenced by automatic metrics and human evaluations.
- We conduct extensive experiments on multiple long-text datasets (such as academic articles and books), and provide in-depth analysis of the results. We examine how the chain-of-thought process impacts summary quality, analyze the trade-offs of hierarchical segmentation, and outline insights that can inform future LLM-based summarization research.

2 Related work

2.1 Overview of text summarization tasks

Automatic Text Summarization (ATS) aims to condense large documents into concise and informative summaries, significantly reducing the cognitive load for users who manage extensive textual data (Zhang et al 2024a, b). Generally, text summarization methods are classified into three paradigms: extractive, abstractive, and hybrid summarization (Zhang et al 2024b; Langston and Ashford 2024; Deroy et al 2024).

Extractive summarization identifies and selects key sentences or phrases directly from the source text to create summaries without altering the original wording (Wibawa et al 2024). Traditional methods such as TF-IDF, Latent Semantic Analysis (LSA), and sentence ranking algorithms, including TextRank and LexRank, represent classic extractive techniques. Such methods offer computational efficiency and factual consistency due to their straightforward extraction process; however, they often yield summaries that lack coherence (Ji et al 2023; Gupta and Gupta 2019). In contrast, abstractive summarization generates new sentences by rephrasing or synthesizing content based on the semantic understanding of the source text (Zhang et al 2022). Earlier abstractive methods relied on structured representations like templates, ontologies, and discourse trees to guide summary generation (Balachandran et al 2020; Xiao et al 2021; Di Noia et al 2018). With the rise of deep learning, neural abstractive models leveraging sequence-to-sequence (seq2seq) architectures, particularly recurrent neural networks (RNNs), long short-term memory (LSTM), and gated recurrent units (GRUs), have significantly improved summarization quality (Chen and Yang 2020; Shini and Kumar 2021; Gillioz et al 2020). However, these neural approaches frequently suffer from issues such as semantic hallucinations and factual inconsistencies (Cao et al 2021; Shakil et al 2024). Hybrid summarization integrates the strengths of both extractive and abstractive methods to enhance summarization quality. Typically, hybrid models first extract salient text segments and subsequently rephrase or paraphrase them into fluent and coherent summaries. Recent studies confirm that hybrid methods, which blend extractive reliability with abstractive flexibility, are highly effective in specialized fields such as healthcare and legal judgments (Kirmani et al 2019; Mahajani et al 2019).

2.2 Large language models and chain-of-thought

Recent advances in Large Language Models (LLMs) such as GPT-3, GPT-4, ChatGPT, and LLaMA have brought substantial improvements to the summarization task (Brown et al 2020; Xu et al 2024; Van Veen et al 2023). These models, pretrained on extensive textual corpora, exhibit remarkable generative capabilities, enabling zero-shot and few-shot summarization via prompt-based learning strategies (Kojima et al 2022; Liu et al 2023). The paradigm flexibility of LLMs allows seamless transitions between extractive, abstractive, and hybrid approaches, greatly improving coherence, fluency, and overall summary quality compared to traditional summarization methods (Zhang et al 2024; Shi et al 2023; Wang et al 2023).

Despite these advantages, LLMs also present significant limitations, particularly in the accuracy of generated content. A persistent challenge is the occurrence of "hallucinations," wherein models generate non-factual or unsupported statements (Ge et al 2023; Perković et al 2024). Such hallucinations pose critical risks in specialized fields like law and medicine, necessitating strategies to enhance model reliability and consistency (Reddy et al 2024; Huang et al 2023).

To address these limitations, Chain-of-Thought (CoT) prompting has emerged as an effective method in natural language processing tasks, particularly complex reasoning and summarization (Zhang et al 2022; Liu et al 2023). CoT prompting guides LLMs to generate intermediate reasoning steps before producing the final output, improving their logical coherence and factual correctness (Yu et al 2024). Recent studies demonstrate that CoT significantly reduces hallucinations by explicitly structuring reasoning processes, thus enhancing model outputs in summarization, mathematical reasoning, and question answering (Sun et al 2025; Zhang et al 2025). CoT-driven summarization shows promising results in preserving key details, improving information completeness, and reducing factual inaccuracies in generated summaries (Lee et al 2024; Yao et al 2024).

2.3 Hierarchical structures in summarization

Hierarchical modeling has proven essential in addressing the inherent complexity of processing and summarizing long documents. Traditional neural summarization models often struggle with capturing extensive context and global structure, leading to information loss and coherence degradation (Wu et al 2021a, b). To mitigate these challenges, hierarchical summarization methods have been proposed, utilizing layered or multi-level encoders and decoders to effectively represent textual information across various granularity levels (Wang et al 2024; Tan et al 2024).

For example, hierarchical encoder-decoder architectures have been successfully applied in summarizing scientific papers and lengthy reports, where lower-level encoders capture local sentence-level semantics, and higher-level encoders aggregate global document structures (Xu et al 2019a, b). Recent work indicates that these hierarchical methods significantly improve summarization quality by capturing long-range dependencies, ensuring thematic con-

sistency, and maintaining structural coherence throughout lengthy documents (Scholes et al 2017; Salam et al 2024).

However, existing hierarchical summarization approaches still face substantial challenges. Despite considerable progress, existing hierarchical models often struggle with domain adaptability and handling large-scale datasets in real-world scenarios. Many such methods assume fixed segmentations or rely on coarse heuristics, which inadequately capture nuanced semantics in complex documents (Wylie and Tregellas 2025; Gao et al 2025). Moreover, lacking a multi-step reasoning mechanism leads to disconnected or semantically inconsistent summaries when texts are extremely lengthy or domain-specific (Irvin et al 2025). By contrast, our proposed approach integrates chain-of-thought prompting within a hierarchical framework, enabling more robust factual consistency and thematic coherence across large-scale and specialized corpora.

Given these unresolved issues, there is a clear necessity for methods that integrate sophisticated reasoning mechanisms, like CoT, with hierarchical input and output modeling. The proposed method in this study addresses this gap by explicitly guiding the LLM through a multi-step reasoning process to enhance consistency and accuracy while using hierarchical structures to efficiently manage long-document contexts. This approach intends to mitigate prevalent issues such as information omission, semantic incoherence, and factual hallucinations, offering a robust framework for long-text summarization.

3 Methods

3.1 Task definition

We address the long-document abstractive summarization problem. Formally, the input is a long document D (e.g., an article or report) consisting of a sequence of text units (sentences, paragraphs, etc.), and the goal is to produce a concise summary S that preserves the most important information from D in significantly shorter form. We denote the length of D (in tokens or words) as $|D|$, which can be very large (often exceeding the context capacity of standard LLMs), while the summary S is much shorter (e.g., $|S| \ll |D|$). The task assumes no information loss of key content: all major points in D should be covered by S under length constraints.

To manage very long D, we consider a hierarchical representation. Let D be segmented into a sequence of m coherent segments as

$$D = [d_1, d_2, \ldots, d_m] \tag{1}$$

Each d_i is a subdocument (e.g., a section or paragraph), and

$$\sum_{i=1}^{m} |d_i| = |D| \tag{2}$$

We do not assume any overlap between segments, and the segments maintain the original order of D. Segmenting the input will allow processing piecewise while respecting the logical structure of the document.

In addition to input/output, we introduce a sequence of intermediate reasoning steps Z to assist summarization. Here Z represents the chain of thought (CoT) that an LLM might generate when solving the task. Each $z \in Z$ is a piece of intermediate text such as a comment, explanation, or sub-summary that helps reason out the final summary. In our method, Z may include, for example, lists of key points from each segment or an outline of D's content. The final summary S will be derived based on these intermediate steps. By explicitly modeling a chain-of-thought, we allow the LLM to generate and utilize rationale rather than producing S in a single shot. We treat the LLM as a black-box language generator f_θ, without modifying its internal architecture. The LLM is only guided through prompts and task structuring.

To summarize, D is the full document (with segments d_i), Z is the sequence of intermediate reasoning outputs (the CoT steps), and S is the final summary. We aim to learn a mapping $D \rightarrow S$ that can be factored through Z. Conceptually, the generation can be viewed as a two-stage process: first produce a reasoning chain Z given D, then produce S given both D and Z. In other words, we factorize the conditional generation as:

$$P_\theta(S \mid D) \approx \sum_{Z} P_0(Z \mid D)\, P_\theta(S \mid D, Z) \tag{3}$$

where P_θ is the probability distribution of the LLM with parameters θ. Our prompting strategy will explicitly instantiate one plausible Z (via the CoT prompt) and then condition on it to generate S. By structuring the task in this way, we leverage the LLM's ability to generate a series of intermediate reasoning steps that lead to a better summary. Figure 2 illustrates the overall framework, and Algorithm 1 provides a step-by-step pseudocode of the entire pipeline.

3.2 Overall architecture

Figure 1 provides an overview of the proposed summarization pipeline, which we call Hierarchical CoT Summarization. The architecture is composed of three main stages: (1) Hierarchical Input Segmentation, (2) Chain-of-Thought Reasoning, and (3) Structure-Guided Summary Decoding. We

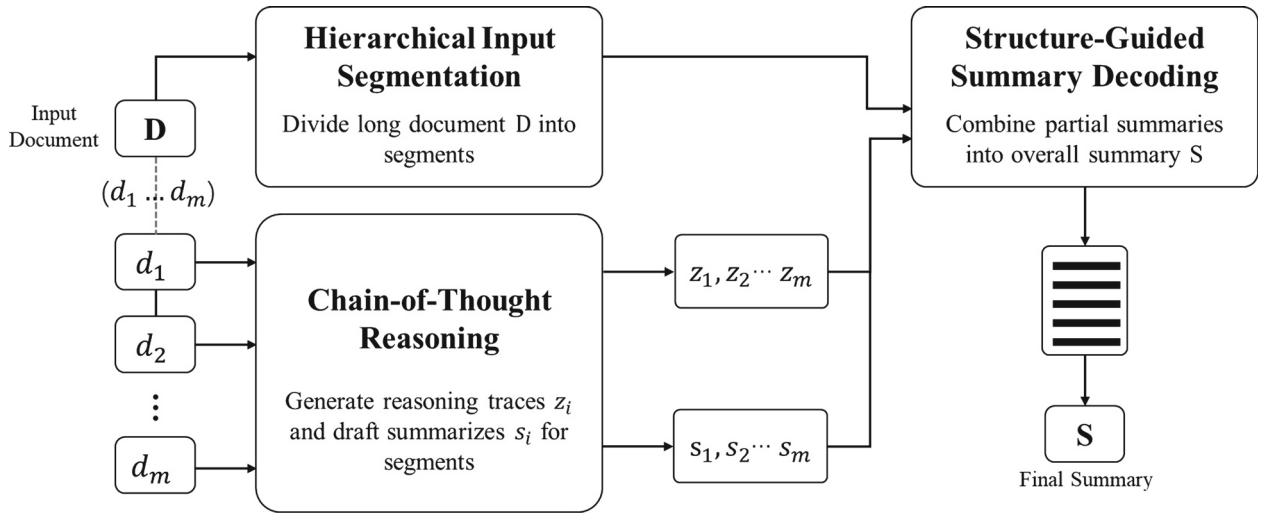

Fig. 1 Overall Architecture of CoTHSSum Framework

Algorithm 1 Hierarchical Summarization with Chain-of-Thought Prompting.

Require: Long document D
Ensure: Final summary S

1: **Segmentation:** Divide D into segments $[d_1, d_2, \ldots, d_m]$ using a hierarchical segmentation method (e.g., by paragraphs or topics).
– Ensure each d_i is within the LLM's context length and is topically coherent.
2: **Initialize** list of draft summaries: *draft_list* ← [].
3: **for** each segment d_i **do**
4: (a) Construct a CoT prompt for d_i to elicit reasoning and summary.
5: (b) Obtain the LLM's output for d_i, including reasoning z_i and a draft summary s_i.
6: (c) Append s_i to *draft_list* (retain z_i if needed for outline).
7: **end for**
8: **Aggregation:** Collate the draft summaries $s_1, \ldots, s_m$ in their natural order.
– Optionally, generate a high-level outline Z_{global} from *draft_list* (e.g., list common themes).
9: **Final Prompt:** Prepare a prompt for the LLM to compose the final summary S:
– Include either the collected s_i or the outline Z_{global}, or both.
– Instruct the LLM to produce a coherent overall summary covering all points.
10: **Decoding:** Generate the final summary S using the LLM with structured guidance.
– (Optionally use constrained decoding or generate multiple candidates and re-rank as discussed in Section 3.5.)
11: **Output:** Return S as the summary of the entire document.

describe each stage at a high level here, and detailed methodologies are given in subsequent subsections.

Hierarchical Segmentation In the first stage, the long document D is divided into smaller, manageable units. This can be a multi-level segmentation depending on D's length and inherent structure. The output of this stage is a sequence of segments $[d_1, \ldots, d_m]$, each of which is short enough to be processed by the LLM. This segmentation addresses the context length limitation and also organizes the content by topic or discourse structure.

CoT-Based Reasoning on Segments In the second stage, each segment d_i is processed with an LLM using a chain-of-thought prompting strategy. Instead of immediately asking for a summary of d_i, the prompt encourages the LLM to reason about the segment-for example, to enumerate the key points or summarize incrementally. The LLM will produce an intermediate reasoning trace z_i for segment d_i, and from that either in the same pass or a subsequent pass produce a draft summary s_i for that segment. These s_i can be seen as partial outputs corresponding to different parts of the document.

Structure-Guided Decoding In the final stage, the partial results (the set of $s_1, \ldots, s_m$ and possibly the collected reasoning traces Z) are integrated to generate the overall summary S. We employ a structured prompting approach that guides the LLM to combine the information from all segments. The decoding can be constrained or assisted by the chain-of-thought so that the final summary is coherent and covers all major content from D. For example, the LLM might be prompted with an outline (constructed from the z_i's) and asked to "write a summary that follows this outline." The result is the final summary S which is presented to the user.

The key innovation of this architecture lies in the external orchestration of the task: by hierarchically segmenting the input and decomposing the reasoning process through chain-of-thought prompting, the method enables large language models such as LLaMA, Qwen, or Phi-1 to handle longer texts and more complex content more effectively. This design allows seamless integration of various LLMs through

carefully structured inputs and instructions, without altering their underlying architecture. In the following sections, we detail each component of the approach.

3.3 Hierarchical input structuring

Given a long document D, the first step is to segment it into a hierarchical set of sub-documents that can be processed individually. We perform segmentation in a way that preserves topical coherence within each segment and yields a logical structure across segments. Formally, we produce segments $d_1, d_2, \ldots, d_m$ such that concatenating the segments in order reconstructs the original D. The segmentation can be done at multiple levels of granularity (paragraphs, sections, chapters), but here we describe a two-level hierarchy for simplicity: first into paragraphs (or smaller units), and then grouping those into higher-level sections if needed.

Segmentation strategy We consider several strategies to obtain meaningful segments of D:

- **Paragraph or Heading-Based Splitting:** If the document has natural paragraph breaks or section headings, we use those as initial segment boundaries. Each paragraph becomes a candidate d_i. This preserves the author-provided structure of D. If sections are explicitly marked, we can also treat each section as a higher-level segment containing multiple paragraphs.
- **TextTiling (Topic Segmentation):** For unstructured text, we can apply a classical segmentation algorithm like Hearst's TextTiling. TextTiling computes lexical similarity between adjacent blocks of text to find topic shift boundaries. It segments D into multi-paragraph passages that each focus on a subtopic. This yields segments that are topically coherent, aligning with how a human might partition the text by subject.
- **Semantic Clustering:** As an alternative or additional step, we can embed each paragraph into a vector space and cluster them to identify topical groups. For example, let $\mathbf{h}_i$ be an embedding of paragraph d_i (obtained by averaging word embeddings or using a sentence transformer). We compute similarities $\cos(\mathbf{h}_i, \mathbf{h}_j)$ between. A clustering algorithm (e.g., agglomerative clustering or k-means) groups paragraphs into k clusters $T_1, \ldots, T_k$ such that each cluster represents a broader topic. Each topic cluster T_j can then be treated as a higher-level segment consisting of a set of related paragraphs. This forms a hierarchical tree: paragraphs grouped into topics, topics possibly grouped into bigger themes, etc.

By these means, the document is hierarchically structured: $D \rightarrow$ sections/topics $\rightarrow$ paragraphs. In our notation, after segmentation we have segments d_i that are ready for summarization. Each segment d_i should ideally be within the token limit of the LLM (e.g., a few hundred words) so that it can be fed entirely into the model. We denote $n_i = |d_i|$ as the length of segment i. A constraint is that n_i is small enough for the LLM context window (for instance, $n_i < 2048$ tokens if the model's context limit is 2048). We also note that the number of segments m will typically be such that $m \times n_i > |D|$, i.e., we expand the document into multiple calls.

Segment representation To later guide the summarization, we may compute a representation for each segment. For example, we can define

$$\mathbf{h}_i = \frac{1}{|d_i|} \sum_{w \in d_i} \mathbf{v}(w) \tag{4}$$

where $\mathbf{v}(w)$ is an embedding vector for word w. Here $\mathbf{h}_i$ is a simple average embedding (bag-of-words representation) of segment d_i. More advanced representations could use pre-trained encoder models (e.g., using the [CLS] token output of BERT on d_i). These vectors can be used to decide clustering as mentioned, or to compute importance weights for each segment (longer or content-heavy segments might be deemed more important for the final summary). However, even without explicit vectorization, the segmentation itself yields an ordered list of segments capturing the document's structure. We will use this structure to ensure all parts of D are summarized. B

y the end of this stage, we have D split into m segments $d_1 \ldots d_m$ (and possibly a higher-level grouping among them). The method now proceeds to process each segment via the LLM with chain-of-thought prompting.

3.4 Chain-of-Thought Reasoning for Summarization

Using the LLM on each segment, we incorporate Chain-of-Thought (CoT) prompting to improve the quality and faithfulness of the summaries. Chain-of-thought prompting means that instead of directly asking the model for the answer (here, a summary of the segment), we ask it to produce a sequence of intermediate reasoning steps that lead to the answer. This technique has been shown to elicit the latent reasoning capabilities of LLMs and improve performance on complex tasks by breaking them down into simpler steps. We adapt this idea to summarization: the model is prompted

to think step-by-step about the content of d_i before finalizing its summary.

CoT Prompt Design For each segment d_i, we construct a prompt that includes an instruction for chain-of-thought. For example, a zero-shot prompt might be:

Prompt Example

```
Prompt for segment d_i:
You are an expert summarizer.
Summarize the following text step
by step.
First, list the key points from
the text. Then provide a concise
summary.
Text: {d_i} Chain-of-Thought:
```

By appending a phrase like "Chain-of-Thought:" or "Let's think step by step.", we cue the model to start generating an explanatory decomposition. The model might then output something like a numbered list of key facts or a reasoning narrative about d_i, and finally the summary. We can also include few-shot examples in this prompt (demonstrating how to list points then summarize) to further guide the model. In our experiments, even a simple cue like "Let's think step by step" is effective at triggering multi-step reasoning in the LLM.

Generation of reasoning and draft summary Using the prompt above, the LLM produces an output that consists of two parts: (a) a sequence of intermediate reasoning steps $z_{i,1}, z_{i,2}, \ldots, z_{i,t}$ (e.g., bullet points of important information in d_i), and (b) a draft summary s_i of segment d_i. We ensure the prompt or few-shot examples clearly delineate these parts (for instance, after listing points, the example might say "Summary:" and then give the summary). In this way, the model's output for each segment d_i is structured as z_i (chain-of-thought text) followed by s_i (the summary for that segment). We treat z_i as an interpretable latent variable that can be observed and utilized. Figure 2 (illustrative) shows that for each segment, the LLM engages in an internal reasoning process before finalizing the segment summary.

Mathematically, for each segment d_i, the LLM is performing a conditional generation:

- Reasoning step: $z_i = f_\theta^{\text{CoT}}(d_i)$,
- Summarization step: $s_i = f_\theta^{\text{summ}}(d_i, z_i)$.

where f_θ^{CoT} and f_θ^{summ} indicate the model under different prompts (the first prompt elicits reasoning z_i, and the second uses z_i to produce s_i). In practice, we often combine these by prompting the model to output both in one go, as described above. The chain-of-thought z_i effectively bridges the gap between d_i and s_i, ensuring that the model has explicitly considered the details of d_i. Prior work on summarization has suggested that such step-by-step summarizing (sometimes called "Summary Chain-of-Thought (SumCoT)") helps integrate fine-grained details and reduces omissions. By mimicking a human's process of first noting key points then writing a summary, we aim to improve the faithfulness of each s_i. In addition, CoT reasoning can reduce factual hallucinations and redundancy, since the model is encouraged to ground each step in the source segment.

We present a pseudocode sketch for the CoT-based segment summarization as shown in Algorithm 2.

Algorithm 2 Summarize Segment with CoT.

```
function SummarizeSegmentWithCoT(segment)
    prompt ← format(
    "List the key points of the following
text, then summarize it:\n Reasoning:")
    output ← LLM.generate(prompt)
    (reasoning, draft_summary) ←
split_output(output, separator="Summary:")
    return draft_summary
end function
```

In the above pseudo-code, $LLM.generate()$ denotes generating text from the model given the prompt. The output will contain the reasoning steps and the draft summary; we then split it by the "Summary:" keyword (as defined in the prompt template) to isolate the summary portion s_i. The returned draft_summary corresponds to s_i. We would call this function for each d_i in D.

After this stage, we have a set of intermediate results: $(z_i, s_i)_{i=1}^m$ for all segments. The chain-of-thought steps z_i can be discarded or optionally retained for analysis, but the crucial pieces for the next stage are the draft segment summaries $s_1, \ldots, s_m$. Each s_i is a condensed version of its segment, and together they cover the content of the original D (assuming the chain-of-thought successfully extracted the key points from each part). The final challenge is to combine these pieces into one coherent summary S of the whole document.

3.5 Structure-guided summary decoding

Simply concatenating all segment summaries $s_1 \ldots s_m$ would produce a very rough summary of D, but it would likely be disjointed and redundant. The purpose of this stage is to compose a well-structured final summary S that flows naturally and covers all important information. We achieve this by using the LLM again, this time fed with the collection of partial summaries or a structured outline derived from them. The decoding is guided in the sense that the intermediate structure (the segmentation and the chain-of-thought outputs)

informs the final generation. We outline several techniques we employ for structure-guided decoding:

1. **Prompt-based Merging:** The simplest approach is to feed the LLM all the draft segment summaries $s_1, \ldots, s_m$ in order, and ask it to "Write a coherent summary of the entire document given the above partial summaries." This can be done by constructing a prompt like: "Here are summaries of parts of a document: [s_1] ... [s_m]. Please merge them into a single, cohesive summary of the document." The LLM will then produce a unified summary S. Because the input to this prompt contains all the segment summaries, the model has access to the essential points from each part of D. We have effectively reduced the long-document summarization to a second-stage summarization of a shorter text (the concatenated s_i's). This two-stage summarization (summarize pieces, then summarize the summaries) is a classic hierarchical approach and ensures that all parts of the document influence the final output.
2. **Outline-Guided Generation:** If we also leverage the chain-of-thought traces z_i, we can form a high-level outline of D. For instance, each z_i might begin with a brief statement of the segment's main topic. By compiling those, or by prompting the LLM to generate an outline Z_{global} of the whole document (perhaps by summarizing the summaries s_i in bullet form), we obtain a structured plan for S. We can then prompt the LLM to follow this outline when writing the final summary. For example: "Outline: 1) [topic A]; 2) [topic B]; 3) [topic C]. Now write a summary of the document following this outline." This guided decoding makes the final text well-organized, with each portion of S corresponding to one of the major segments/topics of D. It also helps avoid omission: since the outline was built from all segments, the model is less likely to ignore a segment in the final output.
3. **Constrained Beam Search for Coverage:** To ensure coverage of all segments, we can use constrained decoding techniques during the LLM generation of S. One way is to identify a set of must-have keywords or phrases from each segment summary s_i (for example, names of entities, or unique terms from each segment). We then constrain the beam search such that each of these keywords appears at least once in the generated summary. This can be implemented with lexically-constrained beam search algorithms that enforce certain tokens to appear. Formally, if K_i is a set of key tokens extracted from segment d_i, we impose the constraint $\forall i, \exists w \in K_i : w \in S$. This structured decoding ensures that content from every part of D is mentioned in S. The result is a summary that covers all topics. We must be careful to keep S concise and fluent, so the constraints are chosen for truly critical tokens only.
4. **Weighted Aggregation and Re-ranking:** We can also generate multiple candidate summaries and choose the one that best reflects the structure. For example, we run the LLM decoding n times (or use beam search to get n best sequences) for the final summary. Then we score each candidate $S^{(j)}$ based on how well it aligns with the intermediate summaries. A simple scoring function can be defined by a coverage metric: $\text{Score}(S^{(j)}) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\text{content of } s_i \text{ is present in } S^{(j)}$, where $\mathbf{1}$ is an indicator that checks if $S^{(j)}$ includes or paraphrases the key idea from segment i. We pick the candidate with the highest score. We can further refine the score by weighting segments differently (e.g., longer or more important segments get higher weight w_i). This effectively yields a weighted aggregation of the partial summaries. In practice, we might implement this by asking the LLM itself to rate which summary is most complete (a form of self-reranking), or by computing embedding similarities between $S^{(j)}$ and each d_i. This re-ranking step adds an extra layer of assurance that the chosen S is structurally sound.
5. **Template-Guided Formatting:** In cases where the summary needs to follow a particular format or template, we can design the final prompt to enforce that. For instance, if the user expects the summary to have multiple paragraphs corresponding to sections of D, we can instruct the LLM accordingly: "Write the summary in three paragraphs, first covering [topic1], then [topic2], then [topic3]." This template is derived from the structure we found. The LLM will then output a multi-paragraph summary S with a predetermined structure. This approach is useful for reproducibility, as it yields summaries with consistent organization (especially important in reports or multi-section documents).

Combining these strategies, our method produces the final summary S in a structure-aware manner. The external structure (segments and CoT reasoning) guides the decoding of S to ensure that no significant part of D is overlooked and that the information is presented in a logical order. Notably, all these steps still treat the LLM as an unmodified model-we only craft the inputs and decoding procedure externally. The result is a more reliable summarization process that leverages the power of LLMs while compensating for their input length limits and occasional lapses in coherence or coverage.

4 Experiments

In this section, we systematically evaluate our proposed summarization framework, combining hierarchical segmentation and Chain-of-Thought prompting, against multiple strong baseline models across diverse benchmark datasets.

4.1 Datasets

We conduct extensive experiments on five widely-used long-document summarization datasets: ArXiv, PubMed, GovReport, BookSum, and CAILsfzy. The key statistics of each dataset are summarized in Table 1.

- **ArXiv** (Cohan et al 2018): Scientific papers in computer science from arXiv.org, typically long documents structured into clear sections, averaging around 6,000 tokens per document.
- **PubMed** (Gupta et al 2021): Biomedical research articles from PubMed Central, characterized by extensive length and structured scientific writing.
- **GovReport** (Huang et al 2021): Government reports and policy documents, lengthy (often exceeding 10,000 tokens), formal, and information-dense, suitable for assessing complex information extraction and summarization.
- **BookSum** (Kryściński et al 2021): Summaries of full-length books with diverse topics and complex narrative structures, suitable for testing summarization across very long texts.
- **CAILsfzy** (Dan et al 2023): A Chinese dataset consisting of long legal documents and judgments, providing a distinct language and domain for robust evaluation of generalizability.

4.2 Evaluation Metrics

We evaluate summarization performance using several standard automatic metrics, including ROUGE (ROUGE-1, ROUGE-2, ROUGE-L), BLEU, BERTScore, and FactCC (factual consistency metric). The details of these metrics are as follows:

- **ROUGE** (Lin 2004): ROUGE is a family of metrics widely used to evaluate automatic summaries by comparing them with human-written reference summaries. The basic idea is to measure how much lexical overlap (in terms of n-grams, sequences, or subsequences) exists between the generated summary and the reference.
  ROUGE-1 and ROUGE-2:
  - ROUGE-1 measures the overlap of unigrams (individual words).
  - ROUGE-2 measures the overlap of bigrams (pairs of consecutive words).

  Both ROUGE-1 and ROUGE-2 typically use Precision (P), Recall (R), and F1 scores:

$$\text{Precision} = \frac{\text{Number of overlapping n-grams}}{\text{Total n-grams in the system summary}} \tag{5}$$

$$\text{Recall} = \frac{\text{Number of overlapping n-grams}}{\text{Total n-grams in the reference summary}} \tag{6}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

  Here, the overlapping n-grams refer to those n-grams that appear both in the generated (system) summary and in the reference summary.
  ROUGE-L:
  - ROUGE-L focuses on the Longest Common Subsequence (LCS) between the system summary and the reference summary. This metric accounts for sequence order (though not necessarily consecutiveness) to capture sentence-level structure.

  The LCS-based recall and precision are computed as follows:

Table 1 Dataset Statistics

Datasets	Domain	Avg. Doc Len.	Avg. Sum. Len.	#Train	#Val	#Test
ArXiv	Scientific Papers	5828.873	280.0169	202914	6436	6440
PubMed	Biomedical Articles	3054.911	209.829	117108	6631	6658
GovReport	Government Reports	7891.058	476.7699	17519	973	972
BookSum	Literary Texts	4327.309	363.7273	9600	1484	1431
CAILsfzy	Legal Documents	1455.29	148.54	10824	1350	1350

"Avg. Doc Len." and "Avg. Sum. Len." represent the average lengths (in tokens) of documents and summaries, respectively. "#Train", "#Val", and "#Test" denote the number of samples in training, validation, and test sets

- Let $LCS(S, R)$ be the length of the longest common subsequence between the system summary S and the reference summary R.
- Define:

$$\text{Recall}_{LCS} = \frac{LCS(S, R)}{\text{length}(R)} \tag{8}$$

$$\text{Precision}_{LCS} = \frac{LCS(S, R)}{\text{length}(S)} \tag{9}$$

- The F1 score for ROUGE-L is then:

$$F1_{LCS} = 2 \times \frac{\text{Recall}_{LCS} \times \text{Precision}_{LCS}}{\text{Recall}_{LCS} + \text{Precision}_{LCS}} \tag{10}$$

ROUGE-L is often used to gauge how well the generated summary captures the structure and ordering of the reference summary.

- **BLEU** (Papineni et al 2002): BLEU is a popular metric initially proposed for machine translation but often applied to summarization. It computes the n-gram precision (from 1-gram to 4-gram), combined with a brevity penalty (BP) to penalize overly short outputs.
  The BLEU score is generally formulated as:

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{11}$$

where:

- N is typically 4 (i.e., up to 4-gram).
- w_n is the weight assigned to the n-gram precision term (often 1/4 for equal weighting).
- p_n is the n-gram precision, which is calculated as the ratio of matching n-gram in the system output to the total n-gram in the system output.
- **BP** is used to penalize translations/summaries that are too short compared to a reference length.

A common definition of BP is:

$$\text{BP} = \begin{cases} 1, & \text{if length}(S) > \text{length}(R), \\ \exp\left(1 - \frac{\text{length}(R)}{\text{length}(S)}\right), & \text{if length}(S) \leq \text{length}(R), \end{cases} \tag{12}$$

where $length(S)$ and $length(R)$ denote the total number of tokens in the system summary and reference summary, respectively.

- **BERTScore** (Zhang et al 2019): BERTScore leverages pretrained language models, such as BERT, to measure semantic similarity rather than lexical overlap. By computing contextual embeddings of the generated summary and reference summary, BERTScore calculates token-level semantic similarities. Unlike ROUGE and BLEU, which rely on surface-level lexical matching, BERTScore is sensitive to semantic accuracy and better aligns with human judgments regarding semantic coherence and content accuracy.
- **FactCC** (Ribeiro et al 2022): FactCC explicitly measures the factual consistency of summaries. It aims to detect and quantify whether factual claims made in a generated summary align accurately with the content of the source document. Typically implemented as a classifier-based model, FactCC identifies inconsistencies, misrepresentations, or incorrect statements in generated summaries. This metric is especially critical for summarization applications where factual correctness is essential, such as summarizing news articles or scientific papers.

By combining these metrics, we can comprehensively assess summarization quality across multiple dimensions, including lexical coverage, fluency, semantic coherence, and factual accuracy.

4.3 Baseline models

We compare our method against eight strong baseline models:

- **T5** (Raffel et al 2020): A unified text-to-text transformer trained on diverse NLP tasks. It excels in transfer learning due to task-general pre-training.
- **BART** (Lewis et al 2020): Combines bidirectional encoder and autoregressive decoder, pretrained via denoising objectives, and proven effective in text generation tasks.
- **PEGASUS** (Zhang et al 2020): Specifically designed for abstractive summarization, employing gap sentence generation pretraining objective for better capturing salient information.
- **Qwen7B** (Wang et al 2024): A 7-billion-parameter Chinese-English multilingual LLM, demonstrating strong zero-shot and fine-tuning capabilities for text summarization.
- **LLaMA8B** (Inan et al 2023): Meta's open LLM series with an 8B parameter size, known for efficient fine-tuning and good performance on downstream tasks.
- **Phi7B** (Abouelenin et al 2025): Microsoft's compact and efficient LLM optimized for practical deployment, showing strong capability on summarization with smaller computational footprints.
- **Yi9B** (Young et al 2024): A multilingual generative LLM by 01.AI with 9B parameters, robust in language understanding and summarization across languages.

- **Baichuan7B** (Baichuan 2023): A Chinese-focused open-source LLM achieving excellent summarization results especially in Chinese NLP tasks.
- **Gemma9B** (Team et al 2024): Google's efficient open LLM, optimized for high-quality generation tasks, with a focus on summarization.

4.4 Main experimental results

The summarization performance across datasets is presented comprehensively in Tables 2 and 3.

Tables 2 and 3 presents the performance of various baseline models and our proposed method across five long-text summarization datasets, evaluated using ROUGE, BLEU, BERTScore, and FactCC. Several key observations emerge from the results:

- **Overall Superiority of the Proposed Method**: Across all datasets and evaluation metrics, our method consistently outperforms both traditional encoder-decoder models (T5, BART, PEGASUS) and recent large language models (Qwen7B, LLaMA8B, Phi7B, Yi9B, Baichuan7B, Gemma9B). The improvements are particularly pronounced on longer and more structurally complex documents, such as those in the GovReport, BookSum, and arXiv datasets. For instance, on arXiv, our model achieves a ROUGE-1 of 48.15, a +1.07 absolute gain over the strongest baseline (Baichuan7B), and +11.11 over PEGASUS. Similar gains are observed for ROUGE-2 (+1.14 over Baichuan7B) and ROUGE-L (+0.12). These consistent gains confirm the effectiveness of combining hierarchical input segmentation and Chain-of-Thought prompting in improving content coverage and logical coherence.

Table 2 Experimental Results on ArXiv, PubMed, GovReport dataset

Datasets	Models	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore	FactCC
arXiv	T5	36.04	11.50	24.35	22.45	60.12	78.54
	BART	42.99	14.97	25.29	23.54	62.48	79.64
	PEGASUS	43.06	16.39	27.65	36.76	70.15	80.45
	Qwen7B	44.15	17.03	28.48	48.55	75.45	84.42
	LLaMa8B	46.37	19.14	30.91	50.92	77.91	86.85
	Phi7B	43.86	16.81	28.13	48.04	74.86	84.13
	Yi9B	44.73	17.64	29.29	49.22	76.23	85.41
	Baichuan7B	47.08	20.01	31.44	51.79	78.58	87.63
	Gemma9B	43.92	15.97	27.76	47.31	73.34	82.89
	CoTHSSum	48.15	21.15	31.56	54.66	78.45	87.98
PubMed	T5	44.65	21.54	40.65	36.54	65.48	82.54
	BART	40.74	14.77	21.59	19.54	58.74	78.64
	PEGASUS	45.97	20.15	41.34	37.15	67.54	85.45
	Qwen7B	45.91	22.33	30.8	51.28	73.15	87.15
	LLaMa8B	47.73	24.09	32.64	52.87	75.66	89.38
	Phi7B	43.58	20.88	29.12	49.14	71.32	85.37
	Yi9B	44.93	21.44	29.97	50.33	72.41	86.22
	Baichuan7B	46.64	23.12	31.55	52.19	74.57	88.74
	Gemma9B	43.03	20.21	28.61	48.89	70.89	84.68
	CoTHSSum	52.14	28.46	36.45	56.25	77.36	89.98
GovReport	T5	27.26	8.24	18.61	15.64	50.78	75.12
	BART	24.17	10.97	16.96	13.98	49.45	70.54
	PEGASUS	25.19	6.71	18.24	16.45	52.14	76.73
	Qwen7B	39.89	18.95	21.82	55.11	73.16	86.16
	LLaMa8B	41.47	21.31	23.94	57.33	75.42	87.52
	Phi7B	37.61	16.48	19.76	52.21	70.54	83.42
	Yi9B	38.73	17.65	20.64	53.38	72.03	84.91
	Baichuan7B	41.06	20.14	22.97	56.24	74.21	87.33
	Gemma9B	36.88	15.31	18.91	50.66	69.17	81.95
	CoTHSSum	42.56	21.54	24.36	57.48	76.12	88.64

Table 3 Experimental Results on BookSum and CAILsgzy dataset

Datasets	Models	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore	FactCC
BookSum	T5	39.88	8.01	13.99	12.41	44.62	60.12
	BART	38.71	7.59	13.65	12.18	43.45	58.64
	PEGASUS	36.03	7.23	12.88	10.66	40.65	50.64
	Qwen7B	41.25	11.54	15.46	14.65	48.15	65.12
	LLaMa8B	43.89	14.02	17.84	16.93	50.61	67.42
	Phi7B	38.94	9.12	13.23	12.68	45.36	62.37
	Yi9B	40.11	10.43	14.31	13.57	47.10	64.03
	Baichuan7B	42.36	12.78	16.65	15.42	49.74	66.88
	Gemma9B	40.02	8.95	14.94	12.73	45.89	61.27
	CoTHSSum	44.54	43.58	18.45	18.05	52.64	68.95
CAILsfzy	T5	56.24	35.88	53.25	45.45	72.45	85.65
	BART	49.36	21.55	39.56	34.54	75.64	85.41
	PEGASUS	50.45	22.65	40.56	38.55	78.65	86.98
	Qwen7B	60.15	40.11	58.12	52.16	83.45	90.12
	LLaMa8B	61.88	42.25	59.73	53.89	85.34	91.50
	Phi7B	58.46	38.29	56.03	49.67	81.52	88.40
	Yi9B	59.37	39.11	57.26	50.84	82.60	89.23
	Baichuan7B	60.94	41.43	58.91	53.22	84.77	90.98
	Gemma9B	57.88	37.55	55.72	48.73	80.91	87.65
	CoTHSSum	64.45	45.25	63.54	58.15	86.15	91.54

- **Performance on Domain-Specific Texts:** In PubMed and CAIL-sfzy, which require both domain expertise and high factual consistency, our method demonstrates superior capability in maintaining faithfulness. On PubMed, our model achieves the highest scores in all metrics, including a ROUGE-2 of 28.46 and a BERTScore of 77.36, outperforming Baichuan7B and LLaMA8B by significant margins. Notably, in CAIL-sfzy, a Chinese legal dataset with complex legal reasoning and terminology, our method achieves the best results across all six metrics (e.g., ROUGE-1: 64.45, BLEU: 58.15, FactCC: 91.54), indicating its ability to generalize across languages and domains with minimal degradation in factuality or fluency.
- **Effectiveness on Extremely Long Documents:** The BookSum and GovReport datasets pose significant challenges due to their length and topic diversity. Here, the benefit of hierarchical segmentation becomes evident. On BookSum, our method surpasses all baselines with a ROUGE-1 of 44.54 and BLEU of 18.05, representing improvements of +2.18 and +2.63 respectively over the next-best performing model (Baichuan7B). These gains are attributable to our strategy of breaking the document into manageable segments and enforcing a structured summarization flow through CoT, allowing the model to reason over content incrementally rather than compressing the entire input in one pass.
- **Factual Consistency and Semantic Coverage:** In terms of factuality, measured by FactCC, our method outperforms all baselines in every dataset. For example, in GovReport, it achieves 88.64, surpassing the next-best Baichuan7B (87.33) and LLaMA8B (87.52). Similarly, BERTScore-a proxy for semantic alignment-indicates consistently better semantic faithfulness. These improvements reflect the ability of CoT prompting to encourage explicit reasoning and reduce hallucinations by forcing the LLM to generate intermediate structures before composing the final output.

These results collectively demonstrate that (1) hierarchical segmentation significantly mitigates input-length limitations; (2) CoT prompting improves interpretability and factual reliability; and (3) when combined, they enable LLMs like Qwen7B to outperform larger or more parameter-heavy models without requiring internal architecture modifications. The consistent performance gains across five distinct datasets affirm the generalizability and effectiveness of the proposed approach.

4.5 Ablation studies

We conduct ablation studies specifically on Qwen7B to assess the individual contributions of CoT prompting and hierarchical structuring. In Table 4, the term “Base" refers

Table 4 Ablation Results

Datasets	Models	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore	FactCC
ArXiv	CoTHSSum	48.15	21.15	31.56	54.66	78.45	87.98
	w/o CoT	46.93	19.78	30.14	52.89	77.16	86.32
	w/o H.S.	45.78	18.64	29.35	51.04	76.21	85.01
	Base	44.15	17.03	28.48	48.55	75.45	84.42
PubMed	CoTHSSum	52.14	28.46	36.45	56.25	77.36	89.98
	w/o CoT	50.03	26.85	34.88	54.67	76.1	88.22
	w/o H.S.	48.32	25.17	33.21	52.86	74.92	87.33
	Base	45.91	22.33	30.8	51.28	73.15	87.15
GovReport	CoTHSSum	42.56	21.54	24.36	57.48	76.12	88.64
	w/o CoT	41.37	20.19	23.12	56.23	75.03	87.52
	w/o H.S.	40.56	19.15	22.05	55.45	74.2	86.73
	Base	39.89	18.95	21.82	55.11	73.16	86.16
BookSum	CoTHSSum	44.54	43.58	18.45	18.05	52.64	68.95
	w/o CoT	43.21	38.16	17.23	17.01	50.84	67.42
	w/o H.S.	42.03	32.89	16.02	15.86	49.42	66.21
	Base	41.25	11.54	15.46	14.65	48.15	65.12
CAILsfzy	CoTHSSum	64.45	45.25	63.54	58.15	86.15	91.54
	w/o CoT	62.87	43.03	61.42	56.41	85.03	90.63
	w/o H.S.	61.38	41.89	60.16	55.03	84.14	90.02
	Base	60.15	40.11	58.12	52.16	83.45	90.12

to the plain Qwen7B model that does not include any CoT prompts or hierarchical structuring enhancements. Ablations include:

- **Ablation of CoT Prompting(w/o CoT):** Trained by removing explicit CoT prompts, directly instructing the model to produce summaries without intermediate reasoning steps.
- **Ablation of Hierarchical Structure(w/o H.S.):** The model processes entire documents directly (limited to max context length), without hierarchical segmentation.

Detailed ablation results are shown in Table 4.

To better understand the individual contributions of the two key components in our proposed method-Chain-of-Thought prompting (CoT) and Hierarchical Structuring (H.S.)-we conduct ablation experiments on all five datasets using Qwen7B as the base model. Specifically, we consider three settings: (1) our full method, (2) without CoT prompting (w/o CoT), and (3) without hierarchical structuring (w/o H.S.). We also report the baseline performance (denoted as base) using Qwen7B without any of our proposed enhancements.

- **Impact of Chain-of-Thought Prompting:** Removing the CoT component results in consistent performance drops across all datasets and metrics. For instance, on the arXiv dataset, ROUGE-2 decreases from 21.15 to 19.78, and FactCC drops from 87.98 to 86.32. Similar trends are observed on PubMed (ROUGE-2: 28.46 $\rightarrow$ 26.85; FactCC: 89.98 $\rightarrow$ 88.22) and CAILsfzy (FactCC: 91.54 $\rightarrow$ 90.63). These results affirm that CoT prompting encourages intermediate reasoning, helping the model to better structure and ground its summary content, leading to improved factual consistency and semantic completeness. Notably, the gain in BERTScore is also substantial (e.g., +1.25 on PubMed), indicating enhanced semantic alignment between generated summaries and reference texts when CoT is included.
- **Impact of Hierarchical Structuring:** Ablating the hierarchical input strategy leads to even sharper declines in performance, particularly on datasets with longer or more complex documents such as BookSum and GovReport. On BookSum, removing H.S. causes ROUGE-2 to plummet from 43.58 to 32.89, and BLEU from 18.05 to 15.86. Similarly, on GovReport, ROUGE-1 drops from 42.56 to 40.56 and FactCC from 88.64 to 86.73. These findings confirm that hierarchical segmentation enables the model to overcome context window limitations, while preserving logical content flow and ensuring better coverage of source documents. The hierarchical scheme proves particularly effective in long-input regimes, where flat summarization strategies fail to encode global discourse structure effectively.
- **Combined Effect and Component Synergy:** While both components contribute positively in isolation, their

combination in our full method consistently achieves the highest scores across all datasets and evaluation metrics. For example, on CAILsfzy, our method achieves a ROUGE-L of 63.54, which is +2.12 over the CoT-only model and +3.38 over the H.S.-only model. This synergy between CoT reasoning and hierarchical document decomposition leads to more logically organized, semantically faithful, and factually consistent summaries.

- **Comparison with Base Model:** The performance gap between our base model (Qwen7B without enhancements) and the full method is substantial across all settings. On PubMed, the full method improves ROUGE-2 by +6.13 points and FactCC by +2.83; on arXiv, BLEU improves by +6.11 points. These improvements highlight that neither CoT nor hierarchical strategies are redundant, but rather jointly critical in tackling the challenges of long-text summarization.

This ablation study provides clear empirical evidence that both CoT prompting and hierarchical structuring are vital for achieving high-quality summarization with LLMs. CoT enhances reasoning and factual reliability, while H.S. improves input handling and content coverage. Their combination results in the most robust and generalizable performance across datasets, domains, and languages.

4.6 Zero-shot and few-shot summarization experiments

To further evaluate the generalizability and practical applicability of our proposed CoTHSSum framework, we conduct additional experiments under low-resource scenarios, specifically examining zero-shot and few-shot summarization capabilities.

We evaluate all five datasets under two distinct conditions:

- **Zero-shot setting:** We directly apply the CoTHSSum framework with carefully designed CoT prompts without providing any training examples from the target dataset.Zero-shot setting: We directly apply the CoTHSSum framework with carefully designed CoT prompts without providing any training examples from the target dataset.
- **Few-shot setting**: We provide the model with three carefully selected in-context examples (few-shot demonstrations) for each dataset. These examples include structured summaries representative of the target domain's characteristics.

All experiments utilize Qwen7B as the base LLM, and we compare our proposed CoTHSSum method against a standard zero-shot and few-shot prompting baseline (Qwen7B without hierarchical segmentation and CoT reasoning).

Table 5 reports zero-shot results. In the zero-shot experiments, our CoTHSSum framework demonstrates significant improvements over the baseline Qwen7B across all evaluated datasets. For instance, on arXiv, CoTHSSum achieves a notable increase in ROUGE-1 (from 29.24 to 34.35) and substantial gains in FactCC (from 62.15 to 64.54). Similar trends are observed on other datasets such as PubMed, GovReport, BookSum, and CAILsfzy, where all metrics-including ROUGE-2, ROUGE-L, BLEU, and BERTScore-consistently show better performance. Table X reports these zero-shot results, highlighting the effectiveness of our approach. These findings indicate that the hierarchical decomposition and explicit chain-of-thought reasoning in CoTHSSum effectively capture and reconstruct key information even when no in-context examples are provided.

Table 6 reports few-shot results. The few-shot experimental results further substantiate the superiority of the CoTHSSum method over the standard baseline. With the incorporation of a limited number of in-context demonstrations, CoTHSSum exhibits enhanced adaptability; for example, in the PubMed dataset, the ROUGE-1 score increases from 33.75 (baseline) to 39.24, while FactCC improves from 68.65 to 73.92. Similar improvements are evident in datasets

Table 5 Zero-shot Summarization Results

Datasets	Method	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore	FactCC
arXiv	Qwen7B	29.24	6.34	15.67	35.15	49.45	62.15
	CoTHSSum	34.35	10.75	20.19	36.52	52.28	64.54
PubMed	Qwen7B	31.52	12.15	21.54	35.16	50.52	64.85
	CoTHSSum	36.85	16.47	25.68	39.45	52.45	71.48
GovReport	Qwen7B	24.12	7.56	11.48	39.15	50.54	63.32
	CoTHSSum	29.85	12.64	15.12	44.45	54.15	69.17
BookSum	Qwen7B	27.25	3.64	5.78	3.15	29.15	47.24
	CoTHSSum	31.63	7.15	8.94	5.48	33.16	53.87
CAILsfzy	Qwen7B	45.12	28.75	44.38	38.45	59.65	73.48
	CoTHSSum	51.42	35.02	51.12	42.65	63.45	77.21

such as arXiv, GovReport, BookSum, and CAILsfzy, where the model achieves higher lexical and semantic precision along with improved factual consistency. Table X reports these few-shot results, underscoring how even scarce external guidance can be leveraged to generate higher-quality summaries. These observations suggest that the structured hierarchical prompting combined with chain-of-thought reasoning significantly enhances the summarization process.

Overall, the comprehensive experimental evaluation confirms that our proposed CoTHSSum framework significantly outperforms conventional prompt-based summarization methods in both zero-shot and few-shot scenarios. The innovative strategies of hierarchical input decomposition and explicit reasoning contribute not only to enhanced information coverage and fluency but also to improved factual accuracy across varied and complex datasets. These results underscore the potential of CoTHSSum to generalize effectively, providing a robust foundation for future research on adaptive and domain-transferrable summarization techniques.

4.7 Human evaluation

We conduct human evaluations to provide additional insight into summarization quality. Three NLP expert annotators independently evaluated 100 randomly selected summaries from arXiv and sfzy, scoring each summary on a 1-5 scale in four dimensions:

Evaluation Criteria and Scoring Guidelines

- **Completeness:**
  Score 1: The summary misses most essential points of the source document, offering only scant coverage.
  Score 2: The summary addresses some key elements but neglects many central topics.
  Score 3: The summary provides adequate coverage of the main points, with some minor omissions.
  Score 4: The summary includes most important information from the source text and is nearly comprehensive.
  Score 5: The summary is fully comprehensive, capturing all critical aspects of the source document.
- **Consistency:**
  Score 1: The summary contains major factual errors or contradictions relative to the original text.
  Score 2: Some statements misrepresent the source or conflict with its content.
  Score 3: The summary is mostly aligned with the original text but contains minor inaccuracies.
  Score 4: The summary is faithful to the original, with no discernible factual errors.
  Score 5: The summary is entirely accurate, with every point verified against the source.
- **Fluency:**
  Score 1: The summary is frequently ungrammatical or unnatural, making it difficult to read.
  Score 2: The text includes noticeable grammatical issues but remains intelligible.
  Score 3: The text flows reasonably well, though some awkward phrasing persists.
  Score 4: The summary reads smoothly, with only minor grammatical or stylistic shortcomings.
  Score 5: The summary is polished, natural, and grammatically flawless.
- **Structural Clarity:**
  Score 1: The summary is poorly organized, without a logical flow or coherent structure.
  Score 2: The organization is somewhat unclear, and transitions between points are weak.
  Score 3: The summary demonstrates some logical structure but could be improved in coherence.
  Score 4: The summary is clearly structured, presenting information in a logical sequence.

Table 6 Few-shot Summarization Results

Datasets	Method	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore	FactCC
arXiv	Qwen7B	31.46	8.85	18.12	35.63	49.81	63.15
	CoTHSSum	36.68	13.02	22.91	40.15	53.18	66.15
PubMed	Qwen7B	33.75	14.22	23.17	37.58	51.15	68.65
	CoTHSSum	39.24	19.98	27.87	42.13	54.15	73.92
GovReport	Qwen7B	27.95	10.28	13.59	41.64	52.45	66.45
	CoTHSSum	32.38	14.72	17.98	46.39	56.45	72.65
BookSum	Qwen7B	29.52	5.32	7.25	4.95	31.54	50.45
	CoTHSSum	33.85	9.46	10.88	7.54	35.65	57.12
CAILsfzy	Qwen7B	48.87	32.66	47.72	41.66	62.68	75.05
	CoTHSSum	53.55	38.73	53.88	44.68	64.14	78.46

Score 5: The summary is exceptionally well-structured, with seamless transitions and a clear logical flow.

Annotator Background We recruited three annotators, each with experience in computational linguistics and summarization-related research. All annotators had advanced academic training in natural language processing (at least at the graduate level) and had participated in similar manual evaluations for other NLP tasks. Before beginning the evaluation, the annotators participated in a calibration session, during which they reviewed sample summaries, discussed the scoring scales, and resolved any ambiguities in the criteria definitions.

Disagreement Resolution Each annotator independently scored a randomly selected set of 100 summaries from both the arXiv and sfzy datasets. After all scores were collected, any discrepancies were addressed through the following steps:

- **Review:** Annotators revisited the source document and corresponding summary to confirm the basis for their ratings.
- **Discussion:** In cases of notable disagreement (e.g., a spread of three or more points on the scale), annotators collectively deliberated, referencing the scoring guidelines to justify their evaluations.
- **Final Score Consolidation:** If a consensus could not be reached through discussion, we applied a majority-vote approach. Where a simple majority did not arise (e.g., each annotator gave a different score), we averaged the three scores to obtain a final numeric value, and rounded to the nearest integer if necessary.

This structured evaluation process ensures consistency, rigor, and transparency in our human judgments of summary quality. By providing clear scoring guidelines, relying on trained annotators familiar with NLP tasks, and implementing a systematic approach to disagreement resolution, we aimed to produce reliable evaluations that complement our automatic metrics.

The average scores are reported in Table 7. Across all datasets and dimensions, our proposed method consistently outperforms the Qwen7B baseline and closely approaches the quality of human-written reference summaries. Notably, the largest gains are observed in structural clarity and completeness, which directly reflect the benefits of hierarchical input processing and stepwise Chain-of-Thought reasoning. For example, on GovReport, our method improves structural clarity from 3.65 (Qwen7B) to 4.18, a notable +0.53 gain, indicating stronger coherence in organizing lengthy, formal content. On CAIL-sfzy, a domain-specific legal dataset, our method attains the highest relative improvements over the base model in consistency (+0.47) and fluency (+0.47), further suggesting that explicit reasoning prompts help maintain factual correctness and linguistic quality in complex legal summarization. While the reference summaries unsurprisingly maintain the highest scores across all criteria, the performance of our method narrows this gap substantially, particularly in PubMed and arXiv, where factual density and terminological accuracy are crucial. Human evaluators consistently reported that summaries generated by our method were more informative, logically ordered, and easier to follow than those produced by the base LLM.

The human evaluation confirms and strengthens the conclusions drawn from automatic metrics: our method gener-

Table 7 Human Evaluation Scores (Avg.)

Datasets	Method	Completeness	Consistency	Fluency	Structural Clarity
ArXiv	Reference Summary	4.91	4.88	4.92	4.85
	Qwen7B	3.86	3.65	3.94	3.78
	Our Method	4.38	4.21	4.46	4.32
PubMed	Reference Summary	4.89	4.91	4.87	4.61
	Qwen7B	3.91	3.75	4.05	3.82
	Our Method	4.41	4.28	4.53	4.37
GovReport	Reference Summary	4.85	4.83	4.84	4.82
	Qwen7B	3.77	3.58	3.84	3.65
	Our Method	4.25	4.09	4.33	4.18
BookSum	Reference Summary	4.88	4.85	4.91	4.87
	Qwen7B	3.66	3.43	3.71	3.59
	Our Method	4.12	3.95	4.26	4.15
CAILsfzy	Reference Summary	4.92	4.94	4.95	4.89
	Qwen7B	4.05	3.89	4.18	3.94
	Our Method	4.53	4.36	4.65	4.41

ates summaries that are not only factually and semantically accurate but also well-structured and fluent. These improvements are a direct result of the synergy between hierarchical input segmentation and CoT-based guided generation, both of which enhance the model's interpretability and reliability from a human perspective.

5 Conclusion

In this paper, we presented a novel summarization framework combining hierarchical input structuring and explicit Chain-of-Thought (CoT) prompting to address the challenges inherent in long-text summarization, such as information omission, semantic incoherence, and factual hallucination. Comprehensive experiments across diverse datasets (ArXiv, PubMed, GovReport, BookSum, and the CAILsfzy legal dataset) demonstrated that our approach significantly outperforms state-of-the-art baseline models, including T5, BART, PEGASUS, Qwen, LLaMA, Phi, Yi, Baichuan, and Gemma, in terms of ROUGE, BLEU, BERTScore, and FactCC metrics. Ablation studies highlighted the critical contributions of both hierarchical segmentation and CoT prompting, while human evaluations further validated the superior factual consistency, completeness, fluency, and structural clarity of our generated summaries. Future work will explore refined segmentation techniques, automatic CoT prompting, and extension to cross-lingual and multimodal summarization tasks.

While effective, our approach is limited by computational efficiency and domain adaptability. Future work will refine segmentation, explore automatic CoT prompting, and extend to cross-lingual and multimodal summarization.

Author Contributions Xiaoyong Chen: Methodology, Writing-original draft, Project administration, Data Curation, Software, Formal analysis, Resources. Zhiqiang Chen: Methodology, Funding acquisition, Validation. Shi Cheng: Supervision, Methodology, Resources, Funding acquisition, Writing-review & editing.

Data Availability The datasets used in this manuscript are publicly available datasets. Detailed information about these datasets is provided in Section 4.1 *Dataset* of this manuscript.

Declarations

Competing Interests 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.

References

Abouelenin A, Ashfaq A, Atkinson A, Awadalla H, Bach N, Bao J, Benhaim A, Cai M, Chaudhary V, Chen C, et al (2025) Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. arXiv:2503.01743

Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, Almeida D, Altenschmidt J, Altman S, Anadkat S, et al (2023) Gpt-4 technical report. arXiv:2303.08774

Altmami NI, Menai MEB (2022) Automatic summarization of scientific articles: A survey. J King Saud Univer-Comput Inf Sci 34(4):1011–1028

Baichuan (2023) Baichuan 2: Open large-scale language models. arXiv:2309.10305

Balachandran V, Pagnoni A, Lee JY, Rajagopal D, Carbonell J, Tsvetkov Y (2020) Structsum: Summarization via structured representations. arXiv:2003.00576

Bao T, Zhang H, Zhang C (2025) Enhancing abstractive summarization of scientific papers using structure information. Exp Syst Appl 261:125529

Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901

Cao M, Dong Y, Cheung JCK (2021) Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. arXiv:2109.09784

Chang Y, Lo K, Goyal T, Iyyer M (2023) Booookscore: A systematic exploration of book-length summarization in the era of llms. arXiv:2310.00785

Chen J, Yang D (2020) Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. arXiv:2010.01672

Choi R, Burns F, Lawrence C (2025) End-to-end chart summarization via visual chain-of-thought in vision-language models. arXiv:2502.17589

Cohan A, Dernoncourt F, Kim DS, Bui T, Kim S, Chang W, Goharian N (2018) A discourse-aware attention model for abstractive summarization of long documents. In: Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: human language technologies, Volume 2 (Short Papers), pp 615–621. Association for Computational Linguistics, New Orleans, Louisiana. https://doi.org/10.18653/v1/N18-2097. https://aclanthology.org/N18-2097

Dan J, Hu W, Wang Y (2023) Enhancing legal judgment summarization with integrated semantic and structural information. Artif Intell Law, 1–22

Deroy A, Ghosh K, Ghosh S (2024) Applicability of large language models and generative models for legal case judgement summarization. Artif Intell Law, 1–44

Deroy A, Ghosh K, Ghosh S (2024) Applicability of large language models and generative models for legal case judgement summarization. Artif Intell Law, 1–44

Di Noia T, Magarelli C, Maurino A, Palmonari M, Rula A (2018) Using ontology-based data summarization to develop semantics-aware recommender systems. In: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June, 2018, Proceedings 15, pp 128–144. Springer

Edge D, Trinh H, Cheng N, Bradley J, Chao A, Mody A, Truitt S, Metropolitansky D, Ness RO, Larson J (2024) From local to global: A graph rag approach to query-focused summarization. arXiv:2404.16130

Floridi L, Chiriatti M (2020) Gpt-3: Its nature, scope, limits, and consequences. Minds Mach 30:681–694

Gao Y, Gan Y, Chen Y, Chen Y (2025) Application of large language models to intelligently analyze long construction contract texts. Construct Manage Econ 43(3):226–242

Ge Y, Guo Y, Das S, Al-Garadi MA, Sarker A (2023) Few-shot learning for medical text: A review of advances, trends, and opportunities. J Biomed Inf 144:104458

Gillioz A, Casas J, Mugellini E, Abou Khaled O (2020) Overview of the transformer-based models for nlp tasks. In: 2020 15th Conference on computer science and information systems (FedCSIS), IEEE pp 179–183

Gupta V, Bharti P, Nokhiz P, Karnick H (2021) Sumpubmed: Summarization dataset of pubmed scientific articles. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing: student research workshop, pp 292–303

Gupta S, Gupta SK (2019) Abstractive summarization: An overview of the state of the art. Exp Syst Appl 121:49–65

Huang L, Cao S, Parulian N, Ji H, Wang L (2021) Efficient Attentions for Long Document Summarization

Huang L, Cao S, Parulian N, Ji H, Wang L (2021) Efficient attentions for long document summarization. arXiv:2104.02112

Huang D, Wei Z, Yue A, Zhao X, Chen Z, Li R, Jiang K, Chang B, Zhang Q, Zhang S, et al (2023) Dsqa-llm: Domain-specific intelligent question answering based on large language model. In: International conference on AI-generated content, Springer pp 170–180

Inan H, Upasani K, Chi J, Rungta R, Iyer K, Mao Y, Tontchev M, Hu Q, Fuller B, Testuggine D, et al (2023) Llama guard: Llm-based input-output safeguard for human-ai conversations. arXiv:2312.06674

Irvin M, Cooper W, Hughes E, Morgan J, Hamilton C (2025) Neural contextual reinforcement framework for logical structure language generation. arXiv:2501.11417

Ji Z, Lee N, Frieske R, Yu T, Su D, Xu Y, Ishii E, Bang YJ, Madotto A, Fung P (2023) Survey of hallucination in natural language generation. ACM Comput Surv 55(12):1–38

Kerui Z, Haichao H, Yuxia L (2020) Automatic text summarization on social media. In: Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control, pp 1–5

Khan B, Shah ZA, Usman M, Khan I, Niazi B (2023) Exploring the landscape of automatic text summarization: a comprehensive survey. IEEE Access 11:109819–109840

Kirmani M, Manzoor Hakak N, Mohd M, Mohd M (2019) Hybrid text summarization: a survey. In: Soft computing: theories and applications: proceedings of SoCTA 2017, Springer pp 63–73

Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y (2022) Large language models are zero-shot reasoners. Adv Neural Inf Process Syst 35:22199–22213

Kryściński W, Rajani N, Agarwal D, Xiong C, Radev D (2021) Booksum: A collection of datasets for long-form narrative summarization. arXiv:2105.08209

Langston O, Ashford B (2024) Automated summarization of multiple document abstracts and contents using large language models. Authorea Preprints

Lee G-G, Latif E, Wu X, Liu N, Zhai X (2024) Applying large language models and chain-of-thought for automatic scoring. Comput Educ: Artif Intell 6:100213

Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2020) Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. The 58th Annual Meeting of the Association for Computational Linguistics(ACL 2020). https://doi.org/10.48550/arXiv.1910.13461

Lin C-Y (2004) Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp 74–81

Liu P, Zhang L, Gulla JA (2023) Pre-train, prompt, and recommendation: A comprehensive survey of language modeling paradigm adaptations in recommender systems. Trans Ass Comput Linguist 11:1553–1571

Liu M, Chen D, Li Y, Fang G, Shen Y (2024) Chartthinker: A contextual chain-of-thought approach to optimized chart summarization. arXiv:2403.11236

Liu H, Liu J, Cui L, Teng Z, Duan N, Zhou M, Zhang Y (2023) Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. IEEE/ACM Trans Audio, Speech, Language Process 31:2947–2962

Mahajani A, Pandya V, Maria I, Sharma D (2019) A comprehensive survey on extractive and abstractive techniques for text summarization. Ambient Commun Comput Syst: RACCCS-2018:339–351

Ou L, Lapata M (2025) Context-aware hierarchical merging for long document summarization. arXiv:2502.00977

Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics, pp 311–318

Perković G, Drobnjak A, Botički I (2024) Hallucinations in llms: Understanding and addressing challenges. In: 2024 47th MIPRO ICT and Electronics Convention (MIPRO), IEEE pp 2084–2088

Pu X, Gao M, Wan X (2023) Summarization is (almost) dead. arXiv:2309.09558

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. The J Mach Learn Res 21:5485–5551. https://doi.org/10.48550/arXiv.1910.10683

Reddy GP, Kumar YP, Prakash KP (2024) Hallucinations in large language models (llms). In: 2024 IEEE open conference of electrical, electronic and information sciences (eStream), IEEE pp 1–6

Ribeiro LF, Liu M, Gurevych I, Dreyer M, Bansal M (2022) Factgraph: Evaluating factuality in summarization with semantic graph representations. arXiv:2204.06508

Salam MA, Aldawsari M, Gamal M, Hamed HF, Sweidan S (2024) Msg-ats: Multi-level semantic graph for arabic text summarization. IEEE Access

Scholes GD, Fleming GR, Chen LX, Aspuru-Guzik A, Buchleitner A, Coker DF, Engel GS, Van Grondelle R, Ishizaki A, Jonas DM, et al (2017) Using coherence to enhance function in chemical and biophysical systems. Nature 543(7647):647–656

Shakil H, Farooq A, Kalita J (2024) Abstractive text summarization: State of the art, challenges, and improvements. Neurocomput, 128255

Shini RS, Kumar VA (2021) Recurrent neural network based text summarization techniques by word sequence generation. In: 2021 6th International Conference on Inventive Computation Technologies (ICICT), IEEE pp 1224–1229

Shi Y, Ren P, Wang J, Han B, ValizadehAslani T, Agbavor F, Zhang Y, Hu M, Zhao L, Liang H (2023) Leveraging gpt-4 for food effect summarization to enhance product-specific guidance development via iterative prompting. J Biomed Inf 148:104533

Sun J, Pan Y, Yan X (2025) Improving intermediate reasoning in zero-shot chain-of-thought for large language models with filter supervisor-self correction. Neurocomput 620:129219

Tan Z, Zhong X, Chiu B (2024) Multimodal paper summarization with hierarchical fusion. In: 2024 International Conference on Engineering and Emerging Technologies (ICEET), IEEE pp 1–6

Team G, Mesnard T, Hardin C, Dadashi R, Bhupatiraju S, Pathak S, Sifre L, Rivière M, Kale MS, Love J, et al (2024) Gemma: Open models based on gemini research and technology. arXiv:2403.08295

Tsirmpas D, Gkionis I, Papadopoulos GT, Mademlis I (2024) Neural natural language processing for long texts: A survey on classification and summarization. Eng Appl Artif Intell 133:108231

Van Veen D, Van Uden C, Blankemeier L, Delbrouck J-B, Aali A, Bluethgen C, Pareek A, Polacin M, Reis EP, Seehofnerova A, et al (2023) Clinical text summarization: adapting large language models can outperform human experts. Res square, 3

Wang J, Liu K, Zhang Y, Leng B, Lu J (2023) Recent advances of few-shot learning methods and applications. Sci China Technol Sci 66(4):920–944

Wang P, Bai S, Tan S, Wang S, Fan Z, Bai J, Chen K, Liu X, Wang J, Ge W, et al (2024) Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv:2409.12191

Wang T, Yang C, Zou M, Liang J, Xiang D, Yang W, Wang H, Li J (2024) A study of extractive summarization of long documents incorporating local topic and hierarchical information. Sci Report 14(1):10140

Wang Y, Zhang Z, Wang R (2023) Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. arXiv:2305.13412

Wibawa AP, Kurniawan F, et al (2024) A survey of text summarization: Techniques, evaluation and challenges. Nat Language Process J 7:100070

Wibawa AP, Kurniawan F, et al (2024) A survey of text summarization: Techniques, evaluation and challenges. Natural Language Process J 7:100070

Wu Z, Jain P, Wright M, Mirhoseini A, Gonzalez JE, Stoica I (2021) Representing long-range context for graph neural networks with global attention. Adv Neural Inf Process Syst 34:13266–13279

Wu C, Wu F, Qi T, Huang Y (2021) Hi-transformer: Hierarchical interactive transformer for efficient and effective long document modeling. arXiv:2106.01040

Wylie KP, Tregellas JR (2025) Rootlets hierarchical principal component analysis for revealing nested dependencies in hierarchical data. Math (2227-7390) 13(1)

Xiao W, Huber P, Carenini G (2021) Predicting discourse trees from transformer-based neural summarizers. arXiv:2104.07058

Xu H, Wang Z, Weng X (2019) Scientific literature summarization using document structure and hierarchical attention model. IEEE Access 7:185290–185300

Xu H, Wang Z, Weng X (2019) Scientific literature summarization using document structure and hierarchical attention model. IEEE Access 7:185290–185300

Xu Z, Jain S, Kankanhalli M (2024) Hallucination is inevitable: An innate limitation of large language models. arXiv:2401.11817

Yang X, Li Y, Zhang X, Chen H, Cheng W (2023) Exploring the limits of chatgpt for query or aspect-based text summarization. arXiv:2302.08081

Yao J, Liu Y, Dong Z, Guo M, Hu H, Keutzer K, Du L, Zhou D, Zhang S (2024) Promptcot: Align prompt distribution via adapted chain-of-thought. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7027–7037

Young A, Chen B, Li C, Huang C, Zhang G, Zhang G, Wang G, Li H, Zhu J, Chen J, et al (2024) Yi: Open foundation models by 01. ai. arXiv:2403.04652

Yu F, Zhang H, Tiwari P, Wang B (2024) Natural language reasoning, a survey. ACM Comput Surv 56(12):1–39

Yu S, Gao W, Qin Y, Yang C, Huang R, Chen Y, Lin C (2025) Itersum: Iterative summarization based on document topological structure. Inf Process Manage 62(1):103918

Zhang T, Ladhak F, Durmus E, Liang P, McKeown K, Hashimoto TB (2024) Benchmarking large language models for news summarization. Trans Ass Comput Linguist 12:39–57

Zhang X, Cao J, Wei J, You C, Ding D (2025) Why does your cot prompt (not) work? theoretical analysis of prompt space complexity, its interaction with answer space during cot reasoning with llms: A recurrent perspective. arXiv:2503.10084

Zhang Y, Gao S, Huang Y, Yu Z, Tan K (2024) 3a-cot: an attend-arrange-abstract chain-of-thought for multi-document summarization. Int J Mach Learn Cybern, 1–19

Zhang Y, Jin H, Meng D, Wang J, Tan J (2024) A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. arXiv:2403.02901

Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y (2019) Bertscore: Evaluating text generation with bert. arXiv:1904.09675

Zhang H, Liu X, Zhang J (2023) Extractive summarization via chatgpt for faithful summary generation. arXiv:2304.04193

Zhang H, Yu PS, Zhang J (2024) A systematic survey of text summarization: From statistical methods to large language models. arXiv:2406.11289

Zhang H, Yu PS, Zhang J (2024) A systematic survey of text summarization: From statistical methods to large language models. arXiv:2406.11289

Zhang Z, Zhang A, Li M, Smola A (2022) Automatic chain of thought prompting in large language models. arXiv:2210.03493

Zhang J, Zhao Y, Saleh M, Liu PJ (2020) Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. Int Conf Mach Learn(ICML 2020), 11328–11339. https://doi.org/10.48550/arXiv.1912.08777

Zhang M, Zhou G, Yu W, Huang N, Liu W (2022) A comprehensive survey of abstractive text summarization based on deep learning. Comput Intell Neurosci 2022(1):7132226

Zhu D-H, Xiong Y-J, Zhang J-C, Xie X-J, Xia C-M (2025) Understanding before reasoning: Enhancing chain-of-thought with iterative summarization pre-prompting. arXiv:2501.04341