StrucSum: Graph-Structured Reasoning for Long Document Extractive Summarization with LLMs

Haohan Yuan[1] **Sukhwa Hong**[2] **Haopeng Zhang**[1] *
[1]ALOHA Lab, University of Hawaii at Manoa
[2]University of Hawaii at Hilo
{haohany, sukhwa, haopengz}@hawaii.edu

Abstract

Large language models (LLMs) have shown strong performance in zero-shot summarization, but often struggle to model document structure and identify salient information in long texts. In this work, we introduce StrucSum, a training-free prompting framework that enhances LLM reasoning through sentence-level graph structures. StrucSum injects structural signals into prompts via three targeted strategies: Neighbor-Aware Prompting (NAP) for local context, Centrality-Aware Prompting (CAP) for importance estimation, and Centrality-Guided Masking (CGM) for efficient input reduction. Experiments on ArXiv, PubMed, and Multi-News demonstrate that StrucSum consistently improves both summary quality and factual consistency over unsupervised baselines and vanilla prompting. In particular, on ArXiv, it increases FactCC and SummaC by 19.2% and 8.0% points, demonstrating stronger alignment between summaries and source content. The ablation study shows that the combination of multiple strategies does not yield clear performance gains; therefore, structure-aware prompting with graph-based information represents a promising and underexplored direction for the advancement of zero-shot extractive summarization with LLMs. Our source code is publicly available. [1]

1 Introduction

Despite recent advances in LLM-based extractive summarization (Zhang et al., 2023a), summarizing long and structured documents remains challenging for large language models due to limited context windows and insufficient discourse awareness (Zhang et al., 2024a). As documents increase in length and complexity, effective summarization requires not only identifying salient sentences but also reasoning over discourse structure and capturing long-range dependencies (Zhong and Litman, 2025; Koh et al., 2022), which continue to pose difficulties for existing approaches.

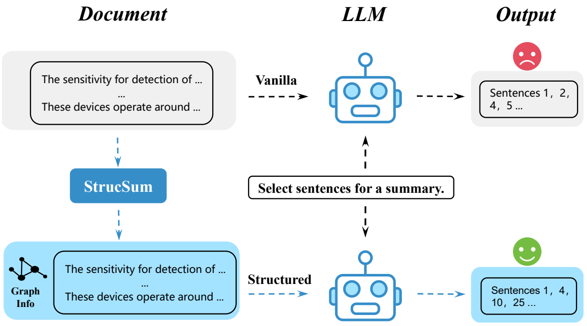
Document
LLM
Output
The sensitivity for detection of ...
...
These devices operate around ...
Vanilla
Sentences 1，2，4，5 ...
StrucSum
Select sentences for a summary.
Graph Info
The sensitivity for detection of ...
...
These devices operate around ...
Structured
Sentences 1，4，10，25 ...

Figure 1: Comparing StrucSum's graph-structured LLM prompting with standard flat-input prompting for long-document extractive summarization.

Extractive summarization selects key sentences to create concise, factually reliable summaries, crucial for domains requiring accuracy and traceability such as science, biomedicine, law, and policy. (Zhang et al., 2023a; Ruan et al., 2022). To capture document-level structure, prior work has explored representing documents as sentence trees or graphs based on semantic, positional, or rhetorical relationships (Mihalcea and Tarau, 2004; Jin et al., 2020; Wang et al., 2020), offering interpretable mechanisms for sentence selection via centrality and coverage (Erkan and Radev, 2004; Cui et al., 2020). However, most existing approaches require supervised training with human written gold summaries (Li et al., 2020; Zhang et al., 2022; Zhao et al., 2024), which limits their scalability and adaptability across domains.

Recently, LLMs have shown strong zero-shot performance in general summarization tasks (Yuan and Zhang, 2024). Their ability to generalize across domains without task-specific fine-tuning makes them attractive alternatives to fully supervised models (Kojima et al., 2023). However, LLMs are trained on unstructured text and often struggle to recognize salient content and maintain factual con-

*corresponding author
[1] https://github.com/HaohanYuan01/StrucSum

Findings of the Association for Computational Linguistics: EACL 2026, pages 3708–3721
March 24-29, 2026 ©2026 Association for Computational Linguistics

sistency in long-form documents due to their lack of structural awareness (Zhang et al., 2024a). Although prompting strategies like in-context learning (Brown et al., 2020) and chain-of-thought (Wei et al., 2023) have enhanced LLM reasoning in other domains, how to incorporate structural reasoning in prompting for long-document summarization remains unexplored.

In this paper, we propose StrucSum, a structure-aware prompting method for long-document extractive summarization in a fully zero-shot setting with LLMs. StrucSum constructs a Text-Attributed Graph (TAG) where sentences are nodes linked by semantic relations, and introduces three strategies to inject structural signals into model inputs: (1) **Neighbor-Aware Prompting** (NAP) adds local sentence context; (2) **Centrality-Aware Prompting** (CAP) appends graph-based importance scores; and (3) **Centrality-Guided Masking** (CGM) removes low-salience content to reduce length. These strategies enhance structural reasoning without any parameter updates. While developed for extractive summarization, StrucSum is model-agnostic and can be readily extended to abstractive or multi-document settings. Across datasets and LLMs, StrucSum improves both summary quality and factuality, revealing that symbolic graph priors can guide black-box LLMs without training. Our main contributions include:

- We propose StrucSum, a structure-aware prompting framework for extractive summarization in a zero-shot, training-free LLM setting, featuring three structural strategies: NAP, CAP, and CGM.

- Experiments on three long-form summarization benchmarks show that StrucSum consistently improves both summary quality and factual consistency over standard prompting.

- We find that different structural strategies support different goals: CGM is most effective for maximizing overlap with reference summaries, NAP best maintains factual consistency, and CAP offers a balance between performance and input length. These findings provide guidance for selecting strategies based on task needs.

2 Related Work

2.1 Graph-based Summarization

Graph-based models are a fundamental approach in extractive summarization, widely used for capturing cross-sentence relationships. Early work, such as TextRank (Mihalcea and Tarau, 2004), constructed sentence graphs based on lexical similarity to rank and select salient content. Subsequent research integrated Graph Neural Networks (GNNs) to learn more expressive, data-driven representations of these document structures (Chhabra et al., 2024). This has led to the development of more complex graph designs for capturing finer-grained semantics, including the bipartite word-sentence graphs in MuchSUM (Mao et al., 2022) and the joint modeling of words and sentences in heterogeneous GNNs (Wang et al., 2020). These techniques have also been adapted for specific contexts: HGSUM (Li et al., 2023) uses a compressed heterogeneous graph for multi-document summarization, G-SEEK (Moro et al., 2023) employs a lightweight selector for low-resource settings, and GSS constructs citation graphs for scientific articles (Chen et al., 2022).

2.2 Summarization with LLMs

Large Language Models have shown strong zero-shot performance on summarization benchmarks like CNN/DailyMail and XSum (Zhang et al., 2024a). Despite these advances, a main challenge is factual consistency, as LLMs are prone to generating hallucinations unfaithful to the source content (Pu et al., 2023). Furthermore, their effectiveness often decreases in specialized domains, such as medical or legal texts (Yuan and Zhang, 2024). To mitigate these issues, QA-prompting (Sinha, 2025) has been introduced to address positional biases by employing question-answering to extract key information before final summary generation. To address domain-specific difficulties, Qi et al. (2025) improves summarization for medical texts by combining prompt optimization and supervised fine-tuning of LLMs. In addition, evaluation frameworks like G-Eval (Liu et al., 2023; Song et al., 2024) further enable finer-grained assessment of summary quality. Although some efforts explore training-free extractive summarization with LLMs (Zhang et al., 2023a), this area remains underdeveloped, particularly for long documents. Our work contributes to improving LLM reasoning in extractive summarization through zero-shot learn-

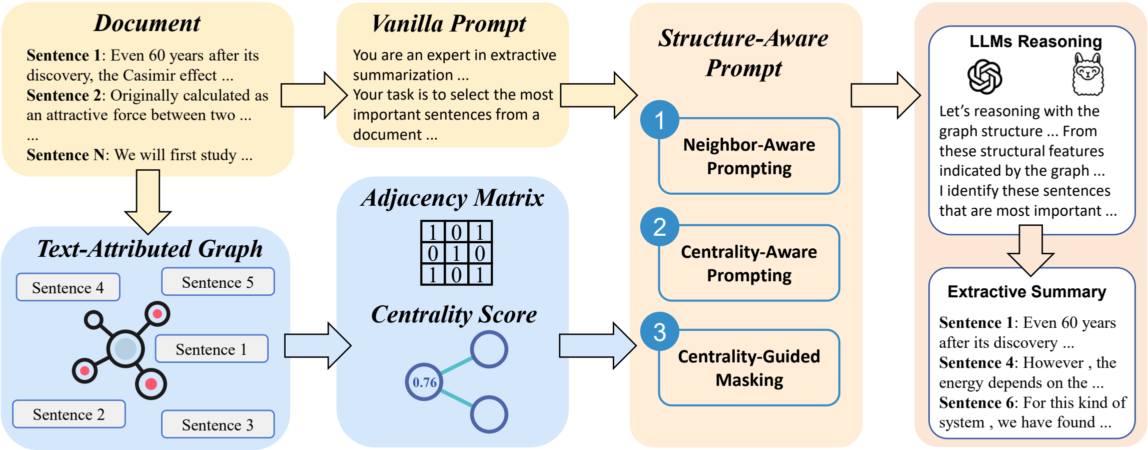
Document
Sentence 1: Even 60 years after its discovery, the Casimir effect ...
Sentence 2: Originally calculated as an attractive force between two ...
...
Sentence N: We will first study ...
Vanilla Prompt
You are an expert in extractive summarization ...
Your task is to select the most important sentences from a document ...
Structure-Aware Prompt
1 Neighbor-Aware Prompting
2 Centrality-Aware Prompting
3 Centrality-Guided Masking
LLMs Reasoning
Let's reasoning with the graph structure ... From these structural features indicated by the graph ...
I identify these sentences that are most important ...
Extractive Summary
Sentence 1: Even 60 years after its discovery ...
Sentence 4: However , the energy depends on the ...
Sentence 6: For this kind of system , we have found ...
Text-Attributed Graph
Sentence 4
Sentence 5
Sentence 1
Sentence 2
Sentence 3
Adjacency Matrix
1 0 1
0 1 0
1 0 1
Centrality Score
0.76

Figure 2: Overview of the StrucSum framework: A document is encoded into a Text-Attributed Graph, from which structural signals are extracted via three prompting strategies to guide LLMs in zero-shot extractive summarization.

ing using structural signals, thereby eliminating the need for additional training or supervision.

3 Method

3.1 Problem Definition

In this work, we model zero-shot extractive summarization as a sentence selection task guided by structural reasoning. Each document is represented as a sequence of N sentences, $\mathbf{S} = (s_1, s_2, \ldots, s_N)$, and encoded as a Text-Attributed Graph (TAG): $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, where each node $v_i \in \mathbf{V}$ corresponds to a sentence s_i, and each edge $(v_i, v_j) \in \mathcal{E}$ captures a semantic or positional relationship between sentence pairs. The objective is to select a subset of k sentences from $\mathbf{S}$ to form the final summary $\hat{y} = \{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\}$.

Our goal is to construct a structured prompt P that integrates both sentence content and graph structure, and use it to query a large language model (LLM) in a training-free setting. We formalize this in two steps:

$$P = GP(\mathbf{S}, \mathcal{G}), \tag{1}$$

$$\hat{\mathcal{S}} = f(P). \quad \text{where } \hat{\mathcal{S}} \subseteq \mathbf{S} \tag{2}$$

Here, $GP(\cdot)$ is a graph-aware prompting function that generates the structured input P from the sentence set $\mathbf{S}$ and its associated graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$. The function $f(\cdot)$ denotes the LLM, which returns a subset of salient sentences $\hat{\mathcal{S}}$ as the extractive summary.

3.2 Graph Construction

To capture inter-sentential relationships, we construct a sparse semantic graph for each document. Specifically, each sentence s_i is encoded using a pre-trained Sentence-BERT model to obtain a dense vector. We compute cosine similarity between all sentence pairs (s_i, s_j) and add an undirected edge $\mathbf{e}_{ij}$ to the edge set $\mathcal{E}$ if $\text{sim}(s_i, s_j) > \theta$, where θ is a tunable similarity threshold. This construction yields a sparse graph structure, which preserves salient relational information while maintaining scalability for long documents.

Finally, this results in a TAG, $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, which encodes both semantic proximity and discourse-level relationships among sentences. These TAGs provide structured signals, such as neighborhood context and sentence importance, integrated into the prompt design, allowing LLMs to reason over document structure during extraction.

3.3 Vanilla Prompting

We begin with a vanilla prompting strategy, which defines the basic mechanism for querying an LLM to perform extractive summarization. This prompt provides the LLM with an explicit task instruction and presents the document as a flat sequence of sentences, without incorporating any structural information. As such, it serves both as a fundamental way to elicit extractive behavior from the LLM and as a baseline for evaluating structure-aware extensions.

Task Description The LLM is instructed to identify and select the k most important sentences from the document. The value of k is a tunable hyperparameter that controls the target summary length.

Prompt Format The vanilla prompt defines the task and represents the document as an ordered list $\mathbf{S} = (s_1, s_2, \ldots, s_N)$. This prompt template is reused and expanded upon in all structure-aware strategies.

> **System Instruction:**
> You are an expert in extractive summarization.
> Your task is to **select the most important sentences** from a document.
>
> **Guideline:** On average, select k key sentences.
>
> Sentence List:
> Sentence 1: "The UN Security Council met to discuss ..."
> Sentence 2: "Several nations expressed concern over ..."
> Sentence 3: "The meeting concluded with a call for ..."
> . . .
> Sentence N: "International observers were invited to ..."
>
> **Expected Output Format:**
> `{ "selected_sentences": [1, 3, 5] }`

3.4 Structure-Aware Prompting

To better guide LLMs in reasoning over document structure, we propose three structure-aware prompting strategies that incorporate both local and global graph signals into the input. These strategies leverage sentence connectivity and importance derived from the TAG, enabling LLMs to make more informed extraction decisions in a fully training-free manner.

Each strategy reflects a different perspective on structural reasoning: (1) **Neighbor-Aware Prompting** integrates local sentence neighborhoods to enhance short-range discourse awareness; (2) **Centrality-Aware Prompting** emphasizes globally important sentences using graph centrality scores; and (3) **Centrality-Guided Masking** filters the input to include only structurally salient content, improving both reasoning focus and input efficiency.

The details of each prompting strategy are as follows:

Neighbor-Aware Prompting (NAP). NAP augments each sentence by explicitly listing its 1-hop neighbors in the TAG, allowing the LLM to consider short-range semantic dependencies and local discourse flow during selection.

The prompt augments each sentence as follows:

> **Context:** Each sentence is followed by its 1-hop neighbors.
>
> **Sentence 1:** "<Text_S1>"
> **Neighbors:** Sentence 2, 6, 7, 9, 13, 14
>
> **Sentence 2:** "<Text_S2>"
> **Neighbors:** Sentence 1, 4, 6, 9, 10, 15
> . . .

By explicitly listing 1-hop neighbors, this format provides the LLM with local contextual cues that highlight semantically related sentences, helping it better assess the role of each sentence within its immediate discourse neighborhood.

Centrality-Aware Prompting (CAP). CAP emphasizes the global structural importance of each sentence by attaching a normalized degree centrality score, computed from the TAG. This score reflects how well-connected a sentence is within the document graph, highlighting key discourse hubs.

The centrality score for sentence s_i is computed as:

$$\text{centrality}(s_i) = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} e_{ij}, \tag{3}$$

where $e_{ij} = 1$ if an edge exists between s_i and s_j, and 0 otherwise.

The prompt format includes each sentence along with its centrality value:

> **Context:** Each sentence is presented with its centrality score.
>
> **Sentence 1 (Centrality: 0.82)**: "<Text_S1>"
> **Sentence 2 (Centrality: 0.51)**: "<Text_S2>"
> . . .

Including centrality values exposes global connectivity patterns, helping the LLM distinguish structurally prominent sentences from peripheral ones.

Centrality-Guided Masking (CGM). CGM introduces structural information by selectively including only a subset of high-centrality sentences in the prompt. Specifically, it ranks sentences by their normalized centrality scores and incrementally selects top-ranked ones until a cumulative centrality threshold ρ (e.g., 85%) is met. The remaining sentences are masked.

The following example shows the resulting prompt:

The full procedure is detailed in Algorithm 1. This approach offers a more efficient encoding of

> **Context:** Only top-ranked sentences (by centrality) are shown in full; others are masked.
>
> **Sentence 1**: "<Text_S1>"
> **Sentence 4**: "<Text_S4>"
> **Sentence 6**: "<Text_S6>"
> ...

structural relevance by reducing input length while preserving key informational content. Full prompt examples for vanilla prompting, NAP, CAP, and CGM are provided in the supplementary materials for clarity and reproducibility.

Algorithm 1: Centrality-Guided Masking (CGM)

```
Input: Sentences $\mathbf{S} = \{s_1, \ldots, s_N\}$; Graph $\mathcal{G}$;
       Threshold $\rho \in (0, 1)$
Output: Prompt $P$ with top-ranked sentences
Compute centrality score $c_i$ for each $s_i$ using $\mathcal{G}$;
Sort $\mathbf{S}$ by $c_i$ in descending order;
Select minimal prefix $\{s_{(1)}, \ldots, s_{(j)}\}$ such that
    $\sum_{k=1}^{j} c_{(k)} \geq \rho \cdot \sum_{i=1}^{N} c_i$;
Build prompt $P$ with full text of selected sentences
    $\{s_{(1)}, \ldots, s_{(j)}\}$;
Mask the remaining sentences;
return $P$;
```

4 Experiment

We conduct a systematic evaluation of our proposed structure-aware prompting strategies: Neighbor-Aware Prompting (NAP), Centrality-Aware Prompting (CAP), and Centrality-Guided Masking (CGM), for zero-shot extractive summarization with Large Language Models (LLMs).

4.1 Experimental Setup

Datasets We conduct experiments on the following three summarization benchmarks:

- **ArXiv** (Cohan et al., 2018): A long-document dataset of scientific papers from diverse domains, each paired with a structured abstract.
- **PubMed** (Cohan et al., 2018): A long-document dataset of biomedical research articles, each accompanied by a concise abstract summarizing complex findings.
- **Multi-News** (Fabbri et al., 2019): A multi-document dataset containing clusters of news articles with human-written summaries that reflect diverse perspectives.

	Arxiv	PubMed	MultiNews
# test	200	200	200
document length	5,745	2,945	1,699
summary length	159	199	219

Table 1: Document and summary lengths (in words) for Arxiv, PubMed, and MultiNews datasets with 200 test samples.

Due to the high inference cost of LLMs, we randomly sample 200 documents per dataset for development and testing. Average document and summary lengths (on the test set) are reported in Table 1.

Large Language Models We experiment with three LLMs: **GPT4o-mini** and **GPT4o** from OpenAI, and **LLaMA3.1-70B-Instruct** from Meta. GPT models are accessed via the official OpenAI API[2], while LLaMA3 is accessed through the NVIDIA API[3]. For calling LLMs, we consistently set the temperature to 0 and used the default "top_p" value of 1.0. The "max_tokens" parameter was set to 100 for all models to control the output length.

Evaluation We evaluate summary quality using a combination of standard, faithfulness-oriented, and holistic metrics. For content overlap, we report ROUGE-2 and ROUGE-L (Lin, 2004), which capture n-gram overlap and longest common subsequence between system and reference summaries. For semantic similarity, we report BERTScore F1 (Zhang et al., 2020), which measures contextual embedding similarity between summaries and references. To assess factual consistency, we use FactCC (Kryściński et al., 2019), a classifier-based metric that detects factual errors in generated summaries, and SummaC (Laban et al., 2022), which estimates sentence-level consistency using natural language inference models. Finally, we report G-Eval (Liu et al., 2023), a GPT-based evaluation framework that provides human-aligned scores across multiple dimensions, including coherence, consistency, fluency, and relevance. G-Eval offers a more comprehensive view of overall summary quality beyond traditional overlap-based metrics.

Unsupervised Baselines We compare against widely-used unsupervised extractive summarization baselines. **Lead** selects the first few sentences as the summary, reflecting positional bias.

[2] https://openai.com/index/openai-api/
[3] https://build.nvidia.com/models

	ArXiv			*PubMed*			*Multi-News*		
System	**R-2**	**R-L**	**BS**	**R-2**	**R-L**	**BS**	**R-2**	**R-L**	**BS**
Unsupervised Systems									
Lead	8.69	26.51	83.10	13.26	30.69	83.88	8.83	25.79	84.81
LexRank (2004)	6.04	22.32	80.94	10.46	28.69	83.13	7.15	23.21	83.64
TextRank (2004)	8.69	23.37	81.63	12.20	29.27	83.29	9.32	25.57	84.35
PacSum (2019)	9.66	26.01	81.97	12.61	31.45	84.46	7.89	24.62	84.08
Large Language Models									
GPT4o-mini	14.14	31.36	84.62	15.66	33.85	84.64	9.46	26.65	84.49
+ NAP	14.94	33.29	84.66	18.03	36.47	84.67	10.13	27.63	84.77
+ CAP	15.39	32.91	84.77	18.09	36.86	84.75	9.96	27.51	84.94
+ CGM	14.99	33.59	84.56	18.21	36.88	84.92	9.84	27.29	84.62
GPT4o	13.74	31.87	84.57	17.53	36.00	85.19	9.49	26.97	84.67
+ NAP	14.63	33.24	84.67	17.87	36.35	85.23	9.89	27.30	84.70
+ CAP	14.60	33.39	0.8478	17.91	36.73	85.25	9.84	27.33	84.69
+ CGM	14.17	32.55	84.21	18.27	36.88	85.25	9.95	27.46	84.62
LLaMA3-70B	13.67	31.32	82.68	16.14	34.21	85.21	10.38	27.87	84.28
+ NAP	13.74	32.09	84.12	16.44	35.30	85.18	10.44	28.22	84.90
+ CAP	13.62	31.69	83.09	16.18	34.72	0.85.14	10.34	27.76	84.74
+ CGM	13.27	31.11	82.53	16.40	34.90	85.25	10.93	28.69	84.64

Table 2: Performance on ArXiv, PubMed, and Multi-News evaluated by ROUGE-2 (R-2), ROUGE-L (R-L), and BERTScore (BS). Best results within each LLM group are highlighted.

LexRank (Erkan and Radev, 2004) builds a similarity graph and ranks sentences via eigenvector centrality. **TextRank** (Mihalcea and Tarau, 2004) uses a PageRank-like algorithm for ranking. **PacSum** (Zheng and Lapata, 2019) incorporates position and polarity to assign edge weights, improving focus in sentence selection.

Implementation Details To construct the TAG, we first encode each sentence using the pre-trained `all-MiniLM-L6-v2` Sentence-BERT model. We then compute cosine similarity between all sentence pairs and add an undirected edge if their similarity exceeds a tunable threshold θ. All hyperparameters are tuned on the development set. The values used for each dataset are provided in Appendix 4.2.

4.2 Hyperparameter Settings

We report the values of three hyperparameters used in our framework. The settings are summarized in Table 3.

Dataset	k	θ	ρ
PubMed	7	0.7	0.8
ArXiv	7	0.6	0.8
Multi-News	9	0.7	0.7

Table 3: Hyperparameter values for number of selected sentences (k), similarity threshold for TAG construction (θ), and centrality coverage ratio in CGM (ρ).

4.3 Summary Quality Evaluation

Table 2 reports the performance of StrucSum and baseline methods on three datasets: ArXiv, PubMed, and Multi-News. We compare unsupervised extractive baselines with LLM-based methods, including both vanilla prompting and structure-aware prompting. We observe that, in most cases, structure-aware prompting outperforms vanilla LLM prompting.

Among the baselines, Lead performs best on PubMed (R-2: 13.26) and remains competitive elsewhere. Graph-based methods like LexRank and PacSum show varying results but fall short of LLM performance, particularly on long documents.

For LLMs, all three structural strategies improve over vanilla prompting. On Multi-News, GPT4o-mini with NAP achieves an R-2 score of 10.13, representing the largest relative gain (+0.67) over its vanilla baseline. On ArXiv, CAP gives the highest R-2 score (15.39) and BERTScore (84.77) among GPT4o-mini variants. On PubMed, CGM performs best, with an R-2 of 18.21 and BERTScore of 84.92.

Overall, structure-aware prompting improves summary quality across different datasets. NAP provides the largest relative improvements, while CAP and CGM achieve the highest absolute scores on specific datasets and metrics.

Dataset	System	FactCC	SummaC	G-Eval
ArXiv	GPT4o-mini	38.44	48.64	4.08
	+ NAP	**57.67**	**56.61**	**4.23**
	+ CAP	45.26	49.07	4.14
	+ CGM	36.19	55.72	4.13
PubMed	GPT4o-mini	44.16	61.89	4.20
	+ NAP	**55.44**	**66.67**	**4.32**
	+ CAP	49.43	65.56	4.25
	+ CGM	45.46	64.94	4.23

Table 4: Evaluation results on summary factual consistency and holistic quality for GPT4o-mini with different prompting strategies. We report scores from two automated factual metrics (FactCC, SummaC) and one general-purpose LLM-based evaluator (G-Eval).

4.4 Summary Faithfulness Evaluation

Extractive summaries can still contain factual inconsistencies, such as coreference errors or misleading sentence combinations (Zhang et al., 2023b). We evaluate summaries generated by GPT4o-mini on ArXiv and PubMed (Table 4) using three metrics: FactCC and SummaC for factual consistency, and G-Eval for summary evaluation with LLM-as-a-judge.

Across both datasets, all structure-aware strategies improve over vanilla prompting. NAP consistently achieves the highest scores on FactCC and SummaC, indicating its strong effectiveness in maintaining factual accuracy. It also slightly outperforms other methods on G-Eval. CAP and CGM show more metric-specific strengths: CAP performs competitively on G-Eval and FactCC, while CGM achieves strong gains on SummaC, particularly on PubMed.

We infer that NAP mainly reduces local reasoning errors rather than increasing reference overlap. By exposing each sentence's 1-hop neighbors, NAP supplies a short-range discourse context that constrains the LLM's sentence stitching. This directly lowers contradiction and attribution mistakes, which FactCC and SummaC are designed to detect, thereby yielding large gains on faithfulness-oriented metrics.

These results suggest that structural prompting enhances the faithfulness of extractive summaries. Among the strategies, NAP is the most effective overall, while CAP and CGM offer complementary benefits depending on the focus of the evaluation.

4.5 Ablation Study of Structural Strategies

We evaluate all valid combinations of NAP, CAP, and CGM on the average of PubMed and ArXiv using GPT-4o-mini. As shown in Table 5, individual strategies generally perform better than combined ones. CGM alone achieves the highest ROUGE score, while NAP alone gives the highest SummaC score. This suggests that CGM is more helpful for improving content coverage, whereas NAP is more effective for consistency. The full combination of NAP, CAP, and CGM performs in the middle for both metrics. In most cases, combining strategies does not lead to further improvement. These results indicate that each structural strategy focuses on different aspects of the summarization task, and using them together does not necessarily lead to better overall performance.

NAP	CAP	CGM	ROUGE	SummaC
✗	✗	✗	0.2781	0.5527
✓	✗	✗	0.3005	**0.6164**
✗	✓	✗	0.2998	0.5732
✗	✗	✓	**0.3034**	0.6033
✓	✓	✗	0.2974	0.5893
✗	✓	✓	0.2979	0.5890
✓	✗	✓	0.3002	0.6160
✓	✓	✓	0.3002	0.5908

Table 5: Ablation and joint analysis of structure-aware strategies on the average of PubMed and ArXiv using GPT4o-mini. ROUGE (average of ROUGE-1/2/L) and SummaC are reported.

Method	Fluent	Informative	Faithful	Overall
GPT4o-mini	2.47	3.81	3.61	3.30
+ NAP	**2.92**	**3.99**	**3.95**	**3.74**
+ CAP	2.57	3.80	3.70	3.38
+ CGM	2.68	3.86	3.79	3.51

Table 6: Human evaluation (1–5 scale) on fluency, informativeness, faithfulness, and overall quality for GPT4o-mini with different prompting strategies.

4.6 Human Evaluation

We conducted a human evaluation on the PubMed dataset using a 1–5 scale for Fluency, Informativeness, Faithfulness, and Overall quality. As shown in Table 6, all proposed strategies (NAP, CAP, CGM) outperform vanilla prompting, with NAP consistently achieving the highest scores across all criteria. The improvements in human-perceived Overall quality (e.g., NAP: 3.74 vs. Vanilla: 3.30) aligns with the gains observed in the holistic G-Eval metric (Table 4). Furthermore, the improved human-rated Faithfulness (e.g., NAP: 3.95 vs. Vanilla: 3.61) is consistent with gains shown by automatic faithfulness metrics like FactCC and SummaC (Table 4). The guidelines for our human evaluators are provided in Appendix B.

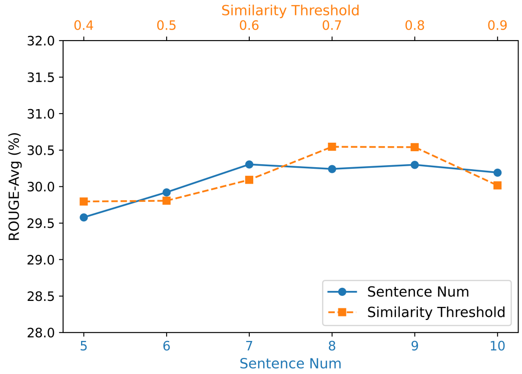

Figure 3: Average ROUGE-F1 on PubMed with varying values of k and θ in the NAP strategy.

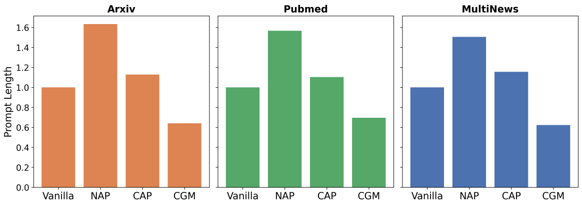

Figure 4: Normalized prompt lengths for vanilla, NAP, CAP, and CGM strategies across ArXiv, PubMed, and Multi-News.

5 Analysis

5.1 Hyperparameter Sensitivity

We evaluate the sensitivity of our proposed methods (we use the Neighbor-Aware Prompting method here as an example, while the other two strategies have shown similar results) to two key hyperparameters: the target number of key sentences (k) and the graph construction similarity threshold (θ). Figure 3 illustrates these results, where different colored lines distinguish the ROUGE-Avg performance curves for each hyperparameter. As shown in Figure 3, the best ROUGE-Avg scores are achieved with approximately 7 key sentences (k) and a similarity threshold (θ) between 0.7 and 0.8. Overall, the results suggest that our method achieves stable performance within a reasonable range of these hyperparameters.

5.2 Token Usage Analysis

Prompt token usage is a critical consideration for practical LLM applications (Sahoo et al., 2025). Figure 4 shows the normalized prompt lengths for our structure-aware strategies: neighbor-aware prompting (NAP), centrality-aware prompting (CAP) and centrality-guided masking (CGM) relative to the vanilla prompting method. Encoding more structural context, NAP and CAP increase input length by approximately 50–60% and 15–18%, respectively. In contrast, CGM, reduces token usage by 40–50% while still showing performance gains over the baseline methods (as detailed in Sections 4.3 and 4.4). Therefore, CGM can be considered as an efficient option for scenarios constrained by input budget or latency.

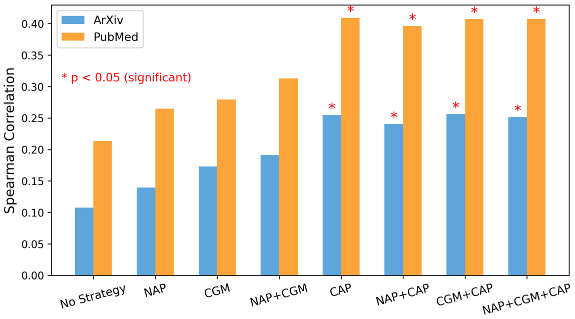

Figure 5: Spearman correlation coefficients between sentence centrality scores and selection rank under different prompting strategies on GPT-4o, shown separately for ArXiv and PubMed. Asterisks indicate statistical significance ($p < 0.05$).

5.3 Correlation Analysis

We assess whether LLMs follow structural cues by computing Spearman correlations between sentence centrality and selection ranks. As shown in Figure 5, the prompts that incorporate centrality (via CAP or combined strategies) produce significant positive correlations ($p < 0.05$), indicating the alignment between selection behavior and importance based on graphs. In contrast, NAP and CGM alone show nonsignificant trends ($p > 0.05$), suggesting a limited effect without explicit importance signals. Although CAP successfully guides LLMs to prioritize central sentences, the results of Table 5 show that CAP-based configurations achieve smaller gains in ROUGE and SummaC compared to NAP and CGM. We infer that this may be because overemphasizing centrality limits the model's flexibility to consider other informative sentences, which can affect both content coverage and consistency.

5.4 Structure-Only Prompt Format Analysis

To evaluate whether LLMs can utilize structural information without access to sentence content, we design a graph-only setting where all textual input is removed and only graph-derived structure

System	*Arxiv* R-2	*Arxiv* R-L	*Arxiv* BS	*PubMed* R-2	*PubMed* R-L	*PubMed* BS
GPT4o-mini						
- TNL	**11.87**	**30.05**	**83.91**	**13.33**	**32.03**	**84.44**
- NAM Matrix	9.50	27.76	83.32	12.75	30.83	84.24
- BAM	9.87	27.78	83.45	12.10	30.21	84.00
+ NAP *	14.59	33.15	84.66	16.42	35.42	85.28

Table 7: Evaluation results on ArXiv and PubMed under the graph-only setting using GPT4o-mini. In this setting, all sentence text is removed and only graph structural information is provided as input. * Indicates results taken from Table 2 .

Dataset	θ	Avg #Nodes	Avg #Edges	Density
ArXiv	0.4	200	7792.6	0.1712
	0.5	200	3197.6	0.0743
	0.6	200	1076.5	0.0260
	0.7	200	304.2	0.0072
	0.8	200	85.6	0.0018
	0.9	200	36.6	0.0007
PubMed	0.4	83	2700.5	0.3011
	0.5	83	1410.3	0.1627
	0.6	83	607.5	0.0744
	0.7	83	210.3	0.0274
	0.8	83	62.6	0.0078
	0.9	83	26.9	0.0027

Table 8: Graph sparsity statistics averaged over 200 documents.

is provided to GPT4o-mini. We compare three prompt formats that encode the same underlying Text-Attributed Graph (TAG) differently: textual neighbor lists (TNL), which list each sentence's neighbors as in the NAP strategy but omit sentence text; numeric adjacency matrices (NAM), dense matrices with cosine similarity scores between sentence pairs; and binary adjacency matrices (BAM), which indicate edge presence based on a similarity threshold θ.

As shown in Table 7, TNL consistently outperforms both NAM and BAM across all metrics. However, its performance remains lower than systems combining structure-aware prompts with full sentence content (for example, NAP achieves 14.59 ROUGE-2 compared to 11.87 for TNL on ArXiv). Overall, these results suggest that in LLM-based summarization, structural cues, although helpful, cannot replace textual information to achieve optimal results.

Graph Sparsity Analysis. We analyze the sparsity of the constructed similarity graphs for the ArXiv and PubMed datasets under different similarity thresholds $\theta \in \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Following standard definitions in network analysis (Newman, 2010), edge density is defined as $|E|/(|V|(|V|-1))$. As shown in Table 8, the resulting graphs are consistently sparse, and their densities decrease rapidly as the threshold increases (e.g., ArXiv: 0.1712 → 0.0007; PubMed: 0.3011 → 0.0027). Our experimental results confirm that the similarity graphs are sparse.

6 Conclusion

We propose StrucSum, a prompting-based framework for long-document extractive summarization that augments LLMs with structural signals from sentence-level graphs. StrucSum applies three strategies: namely, NAP, CAP, and CGM, to guide LLMs in a zero-shot setting. Experiments on three benchmarks show that structure-aware prompting improves both summary quality and factual consistency. StrucSum demonstrates that integrating simple graph priors into prompts enhances LLM reasoning over long documents. Future work may extend this framework to multilingual and dynamic graph reasoning.

Acknowledgment

This work was supported in part by National Science Foundation awards 2201428, 2004014, and 2138259, as well as by computing resources provided through the NSF ACCESS projects CIS250940 and CIS24061. We thank the anonymous reviewers for their valuable suggestions.

Limitations

Instead of conducting experiments on the entire test set, we evaluate a representative subset of 200 test examples per dataset due to budget limits. Previous research efforts have similarly tested LLMs on small subsets (Goyal et al., 2023; Zhang et al., 2024b). As large-scale evaluations remain resource-intensive under current LLM APIs, we leave full-dataset testing to future work.

Our evaluation primarily relies on automatic metrics to assess summary quality and factual consistency. While we include human evaluation for support, its scale is limited. A broader human study, especially involving domain experts, may provide deeper insights.

StrucSum does not involve any fine-tuning, as it is designed for the zero-shot setting. This training-free formulation promotes generality and ease of deployment, though it may limit task-specific performance gains. Exploring fine-tuned variants remains an open direction.

We also restrict our comparisons to vanilla prompting and several unsupervised baselines, without evaluating against supervised or fine-tuned models. This design isolates the effects of structure-aware prompting in a consistent zero-shot context, though broader comparisons could offer further perspective.

Finally, our experiments are conducted in English and on a selected set of LLMs (e.g., GPT-4o, LLaMA3-70B). Evaluating StrucSum across languages, domains, and additional model families is left for future exploration.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Xiuying Chen, Mingzhe Li, Shen Gao, Rui Yan, Xin Gao, and Xiangliang Zhang. 2022. Scientific paper extractive summarization enhanced by citation graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4053–4062, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anshuman Chhabra, Hadi Askari, and Prasant Mohapatra. 2024. Revisiting zero-shot abstractive summarization in the era of large language models from the perspective of position bias. *Preprint*, arXiv:2401.01989.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 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)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Cui, Le Hu, and Yuanchao Liu. 2020. Enhancing extractive text summarization with topic-aware graph neural networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5360–5371, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2023. News summarization and evaluation in the era of gpt-3. *Preprint*, arXiv:2209.12356.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6244–6254, Online. Association for Computational Linguistics.

Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM Comput. Surv.*, 55(8).

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *Preprint*, arXiv:1910.12840.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Miao Li, Jianzhong Qi, and Jey Han Lau. 2023. Compressed heterogeneous graph for abstractive multi-document summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13085–13093.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6232–6243, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *Preprint*, arXiv:2303.16634.

Qianren Mao, Hongdong Zhu, Junnan Liu, Cheng Ji, Hao Peng, Jianxin Li, Lihong Wang, and Zheng Wang. 2022. Muchsum: Multi-channel graph neural network for extractive summarization. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2617–2622, New York, NY, USA. Association for Computing Machinery.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. 2023. Graph-based abstractive summarization of extracted essential knowledge for low-resource scenarios. In *ECAI 2023*, pages 1747–1754. IOS Press.

Mark Newman. 2010. *Networks: An Introduction*. Oxford University Press.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *Preprint*, arXiv:2309.09558.

Kristin Qi, Youxiang Zhu, and Xiaohui Liang. 2025. Umb@peranssumm 2025: Enhancing perspective-aware summarization with prompt optimization and supervised fine-tuning. *Preprint*, arXiv:2503.11118.

Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. Histruct+: Improving extractive text summarization with hierarchical structure information. *Preprint*, arXiv:2203.09629.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2025. A systematic survey of prompt engineering in large language models: Techniques and applications. *Preprint*, arXiv:2402.07927.

Neelabh Sinha. 2025. Qa-prompting: Improving summarization with large language models using question-answering. *Preprint*, arXiv:2505.14347.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024. Finesure: Fine-grained summarization evaluation using llms. *Preprint*, arXiv:2407.00908.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Haohan Yuan and Haopeng Zhang. 2024. Domainsum: A hierarchical benchmark for fine-grained domain shift in abstractive text summarization. *Preprint*, arXiv:2410.15687.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2022. HEGEL: Hypergraph transformer for long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10167–10176, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Extractive summarization via chatgpt for faithful summary generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3270–3278.

Haopeng Zhang, Philip S. Yu, and Jiawei Zhang. 2024a. A systematic survey of text summarization: From statistical methods to large language models. *Preprint*, arXiv:2406.11289.

Shiyue Zhang, David Wan, and Mohit Bansal. 2023b. Extractive is not faithful: An investigation of broad unfaithfulness problems in extractive summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2153–2174, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2024b. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Chenlong Zhao, Xiwen Zhou, Xiaopeng Xie, and Yong Zhang. 2024. Hierarchical attention graph for scientific document summarization in global and local level. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 714–726.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. *arXiv preprint arXiv:1906.03508*.

Yang Zhong and Diane Litman. 2025. Discourse-driven evaluation: Unveiling factual inconsistency in long document summarization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2050–2073, Albuquerque, New Mexico. Association for Computational Linguistics.

A Prompt Examples

In this section, we provide prompt examples for the four main methods evaluated in this paper: Vanilla Prompting and the three structure-aware strategies: NAP, CAP, and CGM.

System Instruction:
You are an expert in extractive summarization.
Your task is to **select the most important sentences** from a document.

Guideline: On average, select k key sentences.

Sentence List:
Sentence 1: “The sensitivity for detecting trace gas molecules in air or other carrier gases by measuring the characteristic vibration–rotation absorption lines in the mid-infrared is far higher than in the near-infrared or visible range.”

Sentence 2: “Thus, highly efficient optoelectronic devices operating in the mid-infrared are essential for sensitive gas analysis and atmospheric pollution monitoring.”

Sentence 3: “Narrow-gap IV–VI semiconductors are promising candidates as mid-infrared detectors, and polycrystalline IV–VI photodetectors operating at room temperature are already commercially available.”

. . .
Sentence 29: “At the low-wavelength edge of the stopband (2.47 μm), an interference fringe appears due to the total thickness of the structure.”

Expected Output Format:

```
{ "selected_sentences": [1, 3, 5] }
```

Vanilla Prompting

System Instruction:
You are an expert in extractive summarization.
Your task is to **select the most important sentences** from a document. Use information about each sentence’s neighboring sentences to better reason about local context.

Guideline: On average, select k key sentences.

Context: Each sentence is followed by its 1-hop neighbors.

Sentence 1: “The sensitivity for detecting trace gas molecules in air or other carrier gases by measuring the characteristic vibration–rotation absorption lines in the mid-infrared is far higher than in the near-infrared or visible range.”
Neighbors: Sentence 2, 4, 7, 10, 13

Sentence 2: “Thus, highly efficient optoelectronic devices operating in the mid-infrared are essential for sensitive gas analysis and atmospheric pollution monitoring.”
Neighbors: Sentence 1, 3, 5, 6, 11

Sentence 3: “Narrow-gap IV–VI semiconductors are promising candidates as mid-infrared detectors, and polycrystalline IV–VI photodetectors operating at room temperature are already commercially available.”
Neighbors: Sentence 2, 4, 6, 8, 12

. . .

Sentence 29: “At the low-wavelength edge of the stopband (2.47 μm), an interference fringe appears due to the total thickness of the structure.”
Neighbors: Sentence 24, 26, 27, 28

Expected Output Format:

```
{ "selected_sentences": [1, 3, 5] }
```

Neighbor-Aware Prompting (NAP)

System Instruction:
You are an expert in extractive summarization.
Your task is to **select the most important sentences** from a document. Use the centrality scores provided to help identify globally important sentences.

Guideline: On average, select k key sentences.

Context: Each sentence is presented with its centrality score.

Sentence 1 (Centrality: 0.82): "The sensitivity for detecting trace gas molecules in air or other carrier gases by measuring the characteristic vibration–rotation absorption lines in the mid-infrared is far higher than in the near-infrared or visible range."

Sentence 2 (Centrality: 0.51): "Thus, highly efficient optoelectronic devices operating in the mid-infrared are essential for sensitive gas analysis and atmospheric pollution monitoring."

Sentence 3 (Centrality: 0.65): "Narrow-gap IV–VI semiconductors are promising candidates as mid-infrared detectors, and polycrystalline IV–VI photodetectors operating at room temperature are already commercially available."

. . .

Sentence 29 (Centrality: 0.47): "At the low-wavelength edge of the stopband (2.47 μm), an interference fringe appears due to the total thickness of the structure."

Expected Output Format:
```
{ "selected_sentences": [1, 3, 5] }
```

Centrality-Aware Prompting (CAP)

System Instruction:
You are an expert in extractive summarization.
Your task is to **select the most important sentences** from a document.
The document has been pre-filtered to include only structurally salient sentences, identified via graph centrality.

Guideline: On average, select k key sentences.

Context: Only top-ranked sentences (by centrality) are shown in full; others are masked.

Sentence 1: "The sensitivity for detecting trace gas molecules in air or other carrier gases by measuring the characteristic vibration–rotation absorption lines in the mid-infrared is far higher than in the near-infrared or visible range."

Sentence 4: "This requirement can be met by using either a narrow-band emitter with a broadband detector, or a broadband source such as a glowbar with a detector of narrow spectral bandwidth."

Sentence 6: "In this so-called resonant cavity enhanced photodetector (RCEPD), a thin absorbing layer is placed in a vertical optical cavity."

. . .

Sentence 29: "At the low-wavelength edge of the stopband (2.47 μm), an interference fringe appears due to the total thickness of the structure."

Expected Output Format:
```
{ "selected_sentences": [1, 3, 5] }
```

Centrality-Guided Masking (CGM)

B Human Evaluation Guidelines

Our human evaluation used two PhD students, each independently rating summaries using the rubric on a 1–5 scale for Fluency, Informativeness, Faithfulness, and Overall Quality. We report the average of their scores across examples. Here we list the full human evaluation guidelines used for our manual annotation study. including rating scales and definitions.

Score	Fluency	Informativeness	Faithfulness	Overall Quality
5	Grammatical, natural, and well-structured; reads smoothly with appropriate register; no typos or awkward phrasing.	Covers all salient points and key details from the source; concise with minimal redundancy; no important omissions.	Fully faithful to the source; no hallucinations, distortions, or unsupported inferences; entity, number, and event details are correct.	Publication-quality summary: accurate, concise, coherent, and follows instructions; no substantive issues.
4	Mostly fluent with only minor issues (rare typos or slightly awkward phrasing) that do not impede comprehension.	Covers most key points, with at most minor omissions or inclusion of unimportant details.	Largely faithful; at most very minor imprecision that does not change meaning or introduce new facts.	High-quality summary with minor issues in at most one dimension; clearly useful as-is.
3	Generally readable but with noticeable issues (grammar/style/word choice) that occasionally hinder the flow.	Conveys several main points but misses multiple important details or includes some irrelevant/redundant content.	Some unsupported statements, paraphrase drift, or partial contradictions; localized factual issues are present.	Adequate but requires edits; there is clear room for improvement in two or more dimensions.
2	Frequent grammatical errors, unclear or choppy sentences; difficult to follow at times.	Captures few key points; omits many important details and/or contains substantial irrelevant content.	Multiple factual errors, contradictions, or hallucinated content; the meaning often diverges from the source.	Low-quality summary with substantial issues; not reliable without significant revision.
1	Severely ungrammatical or incoherent; hard to understand.	Fails to convey the main points; largely irrelevant or off-topic.	Unfaithful to the source, with pervasive hallucinations or contradictions.	Unusable summary that does not meet task requirements.

Table 9: Human evaluation rubric (1–5 scale) for Fluency, Informativeness, Faithfulness, and Overall Quality, used to assess GPT-4o-mini summaries under different prompting strategies.