Beyond the Scientific Document: A Citation-Aware Multi-Granular Summarization Approach with Heterogeneous Graphs

Quoc-An Nguyen, Xuan-Hung Le*, Thi-Minh-Thu Vu*, Hoang-Quynh Le†
VNU University of Engineering and Technology
{annq, 22028172, 22028116, lhquynh}@vnu.edu.vn

Abstract

Scientific summarization remains a challenging task due to the complex characteristics of internal structure and its external relations to other documents. To address this, our proposed model constructs a heterogeneous graph to represent a document and its relevant external citations. This heterogeneous graph enables the model to exploit information across multiple granularities, ranging from fine-grained textual components to the global document structure, and from internal content to external citation context, which facilitates context-aware representations and effectively reduces redundancy. In addition, we develop an effective encoder based on a multi-granularity graph attention mechanism and the triplet loss objective to enhance representation learning performance. Experimental results across three different scenarios consistently demonstrate that our model outperforms existing approaches. Source code is available at: https://github.com/quocanuetcs/CiteHeteroSum.

1 Introduction

Automatic summarization aims to create an abridged version that contains the most critical information from the original text(s) (El-Kassas et al., 2021). As scientific publications are growing at an exponential rate—doubling every nine years—the need for efficient summarization tools is crucial to enhance the productivity of researchers (Bornmann and Mutz, 2015). Unlike abstractive summarization, which generates new sentences, extractive methods select only important information to form a summary. This reduces the risk of hallucination and makes them more reliable for scientific document summarization (Zhang et al., 2023a).

Summarizing scientific texts is challenging due to their complex internal structure. Early graph-based models demonstrated significant potential for adapting to structured data (Erkan and Radev, 2004; Mihalcea and Tarau, 2004). However, these unsupervised models, while effective for structuring, lack the ability to capture deep semantic features without labeled data. The advancement of deep learning models has led to the emergence of sequence-based approaches aimed at capturing cross-sentence relations (Nallapati et al., 2017; Zhou et al., 2018; Beltagy et al., 2020). Recently, graph neural networks have been proposed to combine deep learning with graph architecture, offering a high-level representation of text spans for more effective summarization (Huang and Kurohashi, 2021; Qi et al., 2022; Zhang et al., 2023a; Zhao et al., 2024). Researchers have explored various strategies to optimize graph structures for summarization, from manual section-based input (Cachola et al., 2020) to automatic hierarchical learning (Huang and Kurohashi, 2021; Zhao et al., 2024). However, they primarily focus on internal information while overlooking external relations.

A scientific document is rarely an isolated work; it is part of a larger academic conversation, using references to prior research to discuss or reuse information. As a result, citations have recently been recognized as a valuable source for enhancing summary quality (Yasunaga et al., 2019; Syed et al., 2023). One efficient approach to leveraging

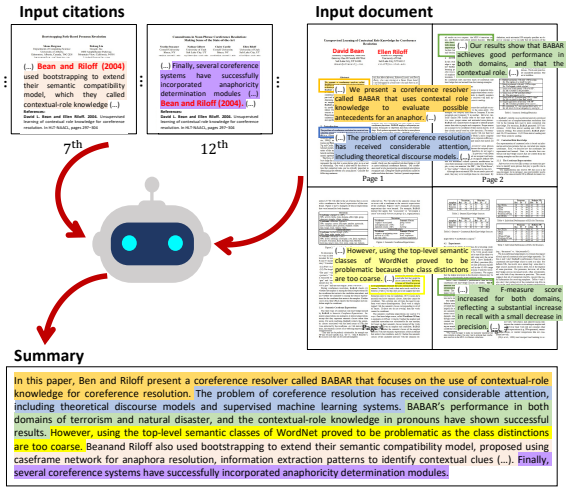

Figure 1: An example of producing a summary from an input document and the citations that reference it.

*Equal contribution
†Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 5034–5046
November 4-9, 2025

citation information is to incorporate short citing spans from external documents that reference the target document, in order to capture the community's perspective and gather concise, up-to-date insights into the paper's contributions (Yasunaga et al., 2019; Syed et al., 2023). Figure 1 shows an example of creating a summary from a document and relevant citations.

In this paper, we propose CiteHeteroSum, a summarization model that creates a summary from multi-granular information, ranging from fine-grained textual components to the global document structure, and from internal content to external citation context. Unlike most existing heterogeneous graph-based models that rely solely on the input document, our model incorporates both the document and relevant citations as input. This dual-input is formatted in a heterogeneous graph before encoding through a multi-granularity graph encoder. Finally, we estimate the importance of each text unit and select the most significant ones to form the summary. Our contributions are:

- We propose a heterogeneous graph architecture that integrates information at varying granularities, enabling context-aware representations and reducing redundancy.
- To learn heterogeneous graph representations and optimize summarization performance, we combine triplet loss and summary loss to guide the learning process and employ a multi-granularity graph encoder to update the model parameters.
- We evaluate our model across different scenarios (high-quality dataset, cross-dataset, and large dataset), demonstrating that it outperforms existing approaches. In addition, we conduct further analysis to highlight the role of the proposed components.

2 Related Work

Text Summarization with Graph Structures Graph-based summarization is a potential approach to model complex relations within texts, as in scientific documents. Early graph-based methods were unsupervised, such as LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004). They built graphs with sentences as nodes and similarities as edges, then applied ranking algorithms to find important content. After that, Approximate Discourse Graph (ADG) (Christensen et al., 2013) was one of the pioneering approaches that leveraged graph structures for training models in text summarization. Subsequent studies have incorporated multiple types of nodes and edges, enabling richer representations of diverse information. For instance, fine-grained units such as words and elementary discourse units (EDUs) enable the elimination of redundant sentence components (Huang and Kurohashi, 2021; Zhao et al., 2024), while sections allow the uncovering of hidden structures within the document (Qi et al., 2022). However, most existing graph-based approaches focus solely on internal content, overlooking valuable external knowledge from citations. In contrast to existing methods, our approach leverages a heterogeneous graph that captures both the internal structure of the document and relevant external citations, providing a more comprehensive representation.

Citation-based Summarization Using citations directly is one of the common approaches to leverage citations in the summarization task (Qazvinian and Radev, 2008; Mohammad et al., 2009; Abu-Jbara and Radev, 2011). However, citation sentences often combine discussions of the input document with references to other works, which can introduce a considerable amount of irrelevant information. Some studies addressed this issue by detecting the cited text spans — portions of the input document most relevant to the citation — and generating summaries based on these spans (Qazvinian et al., 2010; Wang et al., 2017; Agrawal et al., 2019; Yasunaga et al., 2019; Syed et al., 2023). However, the drawback of these approaches is that identifying the correct cited text span is challenging, as a citation may provide an overview of the input document rather than explicitly referring to any specific text span within it. In this study, we propose a new approach that breaks down sentences into elementary discourse units (EDUs) to eliminate irrelevant information in the citation and enables the direct use of citations as input.

3 Methodology

This section formalizes the extractive summarization problem and describes the proposed model. Figure 2 shows the overall model architecture.

Problem Formulation We choose the elementary discourse unit (EDU) as the summary fundamental unit. EDU is commonly used in discourse parsing and is well-suited for structured documents.

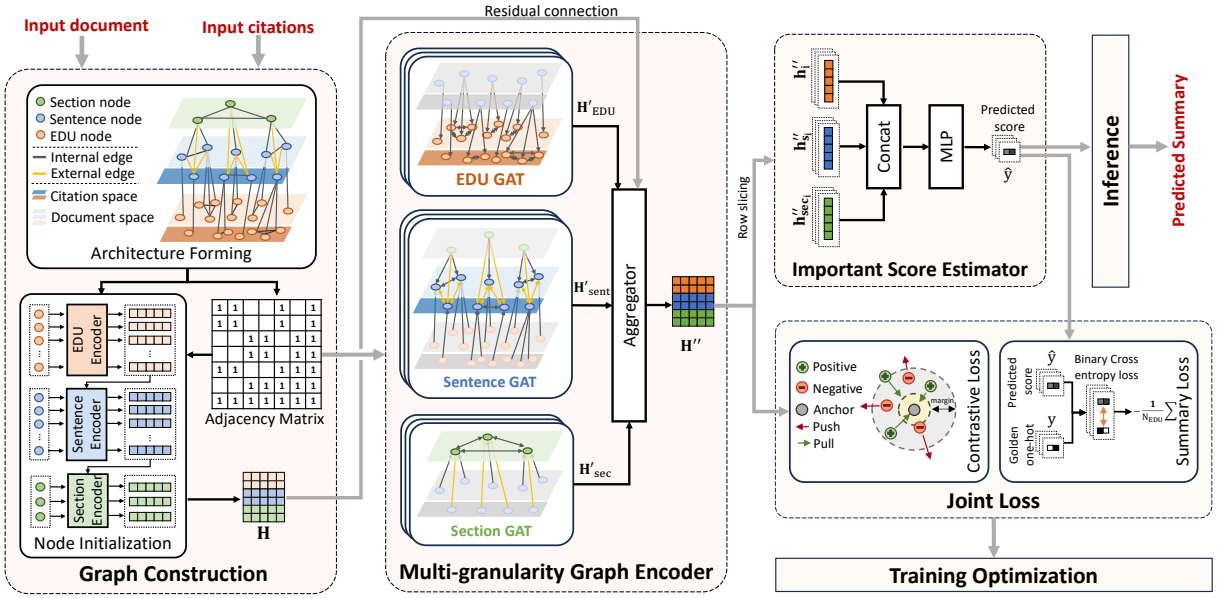

Figure 2: The overall model architecture of the proposed model.

It is a better choice than a sentence because of its finer granularity, especially when a citation may include irrelevant information (Li et al., 2020). Given an input document D containing n_D EDUs and its k citations $\{c_k\}_{k=1}^{n_c}$ where c_k contains n_{c_k} EDUs. The task of the extractive summarization is to predict an important score $y_i \in \{0, 1\}$ for all EDUs, where $y = 1$ represents the i-th EDU that should be included in the summary and $y = 0$ otherwise. The ground truth labels, which we call ORACLE, are extracted following previous work (Qi et al., 2022). During inference, the most important EDUs are selected to form the predicted summary.

3.1 Graph Construction

A document D and its relevant citations $\{c_k\}_{k=1}^{n_c}$ are represented in a heterogeneous graph G.

Architecture Forming We combine node types: EDUs to reduce redundancy, sections to capture structure, and sentences to bridge EDUs and sections in G. A document-level multilingual RST discourse parsing framework is used to extract EDUs (Appendix A). Formally, G has three node types: section (V_{sec}), sentence (V_{sent}), and EDU (V_{EDU}), defined by $V = V_{sec} \cup V_{sent} \cup V_{EDU}$. The edge set E of G is defined as $E = E_{int} \cup E_{ext}$, where E_{int} is the set of internal relations while E_{ext} represents external relations. Formally, an edge E_{int} connects two nodes i and j within a document or a citation if node i is a parent/child of node j or shares the same parent as node j. Meanwhile, E_{ext} integrates a citation into the relevant section by: (1) identifying the most similar section based on cosine similarity, and (2) linking the citation to the found section and the section's sentences.

Node and Edge Initialization We define the initial node representations $\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^{n}$, where n is the number of nodes in G, $\mathbf{h}_i$ is the initial representation of node $i - th$. A node's representation is initialized by aggregating the initialized representation of its children. Firstly, tokens are extracted from each EDU and embedded. Then, the initial representation of EDU e_i is calculated as follows:

$$\mathbf{h}_i = \frac{1}{\mathrm{n}_i} \sum_{j=1}^{\mathrm{n}_i} \mathbf{x}_{t_j} + \mathrm{PE}(p_i) + \mathrm{PE}(p_{s_i}) + \mathrm{PE}(p_{sec_i}) \quad (1)$$

where n_i is the number of tokens in e_i, $\mathbf{x}_{t_j}$ is the vector of token $j - th$, PE(.) is the position encoding function in the Transformer model (Vaswani, 2017) to maintain sequential order information, p_i, p_{s_i} and p_{sec_i} denote the positions of e_i, its sentence and its section, respectively. Similarly, the initial representations of a sentence and a section are computed by averaging the initial representations of their children, without using positional encoding. Edges are represented by an adjacency matrix $\mathbf{A}$, where a relation between two nodes i and j is indicated by a value at positions (i, j) and (j, i). We set the initial relation value between i and j to 1 if node i is connected to node j through E_{int} or E_{ext}, and to 0 otherwise.

3.2 Multi-granularity Graph Encoder

The initial node representation $\mathbf{H}$ and the adjacency matrix $\mathbf{A}$ are fed into the multi-granularity graph encoder to learn node representations.

Graph Attention Network Graph Attention Network (GAT) (Veličković et al., 2017) is used to update node representation after each interaction. When a node i aggregates information from its neighbours, the attention weight is calculated as:

$$\mathbf{z}_{ij} = \text{LeakyReLU}(\mathbf{W}_a[\mathbf{W}_q\mathbf{h}_i||\mathbf{W}_k\mathbf{h}_j]) \quad (2)$$

$$\mathbf{w}_{ij} = \frac{\exp(z_{ij})}{\sum_{k\in\text{Neigh}_i}\exp(z_{ik})} \quad (3)$$

where $\mathbf{W}_a$, $\mathbf{W}_q$, $\mathbf{W}_k$ are trainable weights, Neigh_i is the first-degree neighbors of node $\mathbf{h}_i$, and $||$ is concatenation operation. With multi-head attention, the output is calculated as follows:

$$\mathbf{h'}_i = ||_{k=1}^{\text{K}}\sigma\left(\sum_{j\in\text{Neigh}_i}\mathbf{w}_{ij}^k\mathbf{W}^k\mathbf{h}_j\right) \quad (4)$$

where $||$ is concatenation operation, σ is activation function, $\mathbf{w}_{ij}^k$ are attention weight computed by k^{th} attention mechanism and $\mathbf{W}^k$ is corresponding trainable weight.

Multi-granularity Graph Attention We employ GAT across different node types in the heterogeneous graph to capture information from local to global levels (Zhang et al., 2023a). Firstly, we customize the adjacency matrix at each level by disabling edges that do not connect nodes of that type as follows:

$$\begin{aligned}\mathbf{A}_{\text{EDU}} &= \mathbf{A}\odot\mathbf{1}_{\{i\in\text{EDU}\vee j\in\text{EDU}\}}\\ \mathbf{A}_{\text{sent}} &= \mathbf{A}\odot\mathbf{1}_{\{i\in\text{sent}\vee j\in\text{sent}\}}\\ \mathbf{A}_{\text{sec}} &= \mathbf{A}\odot\mathbf{1}_{\{i\in\text{sec}\vee j\in\text{sec}\}}\end{aligned} \quad (5)$$

where $\mathbf{A}$ is the full adjacency matrix, $\odot$ denotes element-wise multiplication, and $\mathbf{1}_{\{\cdot\}}$ is the indicator function that returns 1 if node i or node j belongs to the specified type.

Each node type — EDU, sentence, and section — is updated separately by GAT as follows:

$$\begin{aligned}\mathbf{H}'_{\text{EDU}} &= \text{GAT}_{\text{EDU}}(\mathbf{H}, \mathbf{A}_{\text{EDU}})\\ \mathbf{H}'_{\text{sent}} &= \text{GAT}_{\text{sent}}(\mathbf{H}, \mathbf{A}_{\text{sent}})\\ \mathbf{H}'_{\text{sec}} &= \text{GAT}_{\text{sec}}(\mathbf{H}, \mathbf{A}_{\text{sec}})\end{aligned} \quad (6)$$

The nodes that have been isolated in $\mathbf{H}$ through the customized adjacency matrices are not updated.

The updated representations are concatenated and passed through a multi-layer perception (MLP) layer to aggregate information across different granularities:

$$\mathbf{H}' = \sigma\left(\mathbf{W}_{\text{agg}}\left[\mathbf{H}'_{\text{sec}} \parallel \mathbf{H}'_{\text{sent}} \parallel \mathbf{H}'_{\text{EDU}}\right]\right) \quad (7)$$

where $\mathbf{W}_{\text{agg}}$ is trainable weight, σ is activation function and $||$ is concatenation operation.

Finally, a residual connection is applied to enhance learning stability and performance.

$$\mathbf{H}'' = \mathbf{H}' + \mathbf{H} \quad (8)$$

Contrastive Learning Loss Contrastive learning loss is employed as one of the objective functions to enhance representation learning. In contrast to prior work using InfoNCE (Zhang et al., 2023a; Nguyen et al., 2025), we adopt triplet loss for its simplicity and greater interoperability (Ostendorff et al., 2022):

$$\mathcal{L}_{\text{c}} = \frac{1}{\text{N}_{\text{t}}}\sum_{i=1}^{\text{N}_{\text{t}}}\max\left\{d(\mathbf{h}_i^a, \mathbf{h}_i^p) - d(\mathbf{h}_i^a, \mathbf{h}_i^n) + \alpha, 0\right\} \quad (9)$$

where N_{t} is the total number of triplets, $\mathbf{h}_i^a$ is the representation of golden summary, $\mathbf{h}_i^p$ is the representation of positive nodes, $\mathbf{h}_i^n$ is the representation of negative nodes, $d(.,.)$ is the Euclidean distance, α is the loss margin.

Because hard samples have been shown to improve model performance (Ostendorff et al., 2022), we propose an approach to select hard samples for our model. Firstly, a margin is used to avoid positive and negative samples colliding:

$$f(e_i) = \begin{cases}\text{Positive} & \text{if ROUGE-2 P}_i \geq \beta\\ \text{Negative} & \text{if ROUGE-2 P}_i \leq \gamma\end{cases} \quad (10)$$

where β and γ are thresholds, $\Delta = \beta - \gamma$ is hard margin, $\Delta > 0$. The model then selects hard positives as the top k EDUs with the lowest scores among positive samples and hard negatives as the top k EDUs with the highest scores among negative samples (Appendix B).

3.3 Important Score Estimation

After the multi-granularity graph encoder phase, we obtain the final EDU node representations $\mathbf{H}'' = \{\mathbf{h}_i''\}_{i=1}^n$. In this phase, the importance of each EDU is estimated, and the final joint loss is computed for training optimization.

Important Score Estimation The representations of EDU $\mathbf{h}''_i$ and its corresponding sentence $\mathbf{h}''_{s_i}$ and section $\mathbf{h}''_{sec_i}$ are concatenated and then fed into MLP layers to predict an importance score:

$$\hat{\mathrm{y}}_{\mathrm{i}} = \varphi(\mathbf{W}_o[\mathbf{h}''_i||\mathbf{h}''_{s_i}||\mathbf{h}''_{sec_i}] + b) \quad (11)$$

where φ is Sigmoid activation function, $\mathbf{W}_o$ is the weight matrix, b is the bias. $\hat{\mathrm{y}}_{\mathrm{i}}$ reflects the importance of the EDU or the likelihood of the EDU being included in the summary.

Summary Loss The summary loss is calculated to optimize the important score of EDUs. By comparing the predicted score $\hat{y}_i$ with the ground-truth label $y_i \in \{0, 1\}$, the loss is defined using binary cross-entropy as follows:

$$\mathcal{L}_{\mathrm{s}} = -\frac{1}{\mathrm{N_E}} \sum_{i=1}^{\mathrm{N_E}} [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \quad (12)$$

where $\mathrm{N_E}$ is the number of EDU.

The final loss combines contrastive and summarization losses as follows:

$$\mathcal{L}_{\text{total}} = \lambda \mathcal{L}_{\text{contrastive}} + (1 - \lambda)\mathcal{L}_{\text{summ}} \quad (13)$$

where λ is a hyperparameter that balances representation learning and summary quality.

Summary Inference To ensure diversity, our model selects EDUs in descending order of importance, skipping any EDU e_i if there exists a previously selected EDU e_j such that ROUGE-2 $\mathrm{P}(e_i, e_j) > \kappa$. EDUs are selected until the desired summary length is reached. Finally, the elected EDUs are reordered according to their original sequence and concatenated to form the predicted summary.

4 Experiments

This section details the experimental design for evaluating the effectiveness of the proposed model.

4.1 Experiment Setup

Datasets and Test Scenarios We used three scenarios to evaluate the proposed model on data with varying sizes and characteristics. Table 1 shows statistics for three scenarios.

Scenario 1: High-quality Dataset We use a high-quality dataset created by human annotators to evaluate the model. The CL-SciSumm dataset, a reliable resource used in the shared task for years, is employed in this scenario (Chandrasekaran et al., 2020). The training set has 40 samples, while the testing set has 62 samples.

Table 1: The dataset statistics for scenarios

	1. High-quality		2. Cross		3. Large	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
Document						
Num	40	62	1000	102	17122	2767
Avg Len	5964	4647	3872	5163	5937	5906
Citation						
Avg/Doc	18.83	15.40	17.37	16.59	8.75	7.51
Avg Len	433	384	428	404	236	271
Summary	Human		Human		Abstract Sec	

Acronym: Num (Number of), Len (Length), Avg (Average), Sec (Section)

Scenario 2: Cross-Dataset We aim to assess the model's generalization by training and testing on different datasets. Model is trained on SciSumNet dataset (Yasunaga et al., 2019) and tested on the full CL-SciSumm dataset (Chandrasekaran et al., 2020). The training set consists of 1000 semi-automatically generated samples, while the test set comprises 102 human-annotated samples.

Scenario 3: Large Dataset To further validate the model's generalization and scalability, we use a large automatically generated dataset, CiteArXiv (Nguyen et al., 2025). Compared to the datasets in Scenarios 1 and 2, the summary labels in CiteArXiv are automatically extracted from the abstract section. This scenario has 17122 training samples and 2767 testing samples.

Evaluation Multiple evaluation metrics are employed, including ROUGE-1, ROUGE-2, ROUGE-L (Lin, 2004), BERTScore (Zhang* et al., 2020), BLEU (Papineni et al., 2002), and METEOR (Banerjee and Lavie, 2005).

Implementation Details The proposed model is trained on an NVIDIA A40. SciBERT is used to generate initial token embeddings (Beltagy et al., 2019). The GAT is configured with 2 layers and a hidden size from 128 to 256. The model is trained using Adam with a learning rate of 0.0003, and a dropout rate of 0.3. The best training-loss scaling factor is set to 0.5. For triplet loss, $\beta = 0.5$, $\gamma = 0.4$ and $k \in \{3, 5\}$ (Appendix C). The version of the model incorporating the EDU layer contains 3.30M parameters, which is a 21.8% increase compared to the version without it. Inference on a document with 11852 tokens takes 0.109 seconds (Appendix D).

Table 2: Experimental Results Across Three Scenarios

Model	R-1	R-2	R-L	BERT	BLEU	Meteor
Scenario 1: High-quality Dataset						
ORACLE*	66.38	60.08	61.04	-	-	-
LEAD	38.37	22.48	28.22	86.72	8.59	19.61
LexRank	33.08	17.52	18.18	83.20	9.46	38.80
PACSUM	34.00	15.12	18.00	83.10	4.26	16.93
$\text{Llama}^{\dagger}_{\text{tiny}}$	10.16	7.01	8.11	84.97	3.02	18.79
$\text{Phi-3}^{\ddagger}_{\text{mini}}$	10.42	7.17	8.21	85.50	3.12	18.70
$\text{Flan-T5}^{\dagger}_{\text{base}}$	51.05	31.17	36.79	87.75	19.33	34.69
$\text{GPT-4o}^{\ddagger}_{\text{mini}}$	50.81	31.03	34.49	87.93	20.99	42.53
PEGASUS	33.97	20.95	27.04	86.14	4.78	16.50
BART	45.01	26.17	32.72	87.22	13.95	26.27
GraphSAGE	32.38	12.14	17.35	82.96	7.16	25.87
GCN	37.32	14.01	20.33	83.86	10.78	30.62
HSG	36.33	33.83	21.43	84.61	7.39	22.42
HAESum	43.20	24.15	29.19	86.70	12.19	24.82
CHANGES	57.60	36.82	41.17	88.48	27.15	46.14
CiteArXiv	57.12	35.79	41.09	88.15	25.90	45.58
Ours	**59.90**	**40.92**	**45.16**	**89.17**	**31.02**	**50.41**
Scenario 2: Cross-Dataset						
ORACLE*	65.40	57.80	59.01	-	-	-
LEAD	36.36	20.86	26.51	86.44	5.80	18.25
LexRank	31.10	16.47	16.74	83.14	8.25	37.08
PACSUM	30.27	14.00	15.59	83.94	3.22	14.51
$\text{Llama}^{\dagger}_{\text{tiny}}$	10.06	7.06	8.04	84.75	3.03	18.84
$\text{Phi-3}^{\ddagger}_{\text{mini}}$	10.24	7.19	8.13	85.22	3.10	18.82
$\text{Flan-T5}^{\dagger}_{\text{base}}$	48.21	27.56	34.45	87.40	16.13	30.52
$\text{GPT-4o}^{\ddagger}_{\text{mini}}$	50.30	28.61	33.06	87.57	19.12	41.20
PEGASUS	33.16	19.40	26.01	85.87	3.94	16.11
BART	43.58	23.83	30.67	86.75	11.20	24.83
GraphSAGE	31.11	10.33	15.87	82.53	5.77	23.34
GCN	36.96	12.97	20.27	84.11	9.87	28.98
HSG	37.47	27.87	28.63	87.35	7.02	21.81
HAESum	44.80	25.67	28.67	87.31	12.84	25.94
CHANGES	52.15	30.45	34.47	87.09	20.51	39.86
CiteArXiv	53.90	32.18	37.56	87.44	22.62	41.44
Ours	**55.67**	**34.73**	**39.09**	**87.96**	**24.76**	**44.17**
Scenario 3: Large Dataset						
ORACLE*	65.03	34.13	43.35	-	-	-
LEAD	29.18	6.91	16.36	82.62	2.29	17.12
LexRank	30.38	6.57	16.43	81.86	1.78	16.63
PACSUM	30.44	7.08	14.96	81.36	3.45	23.23
$\text{Llama}^{\dagger}_{\text{tiny}}$	14.81	1.24	11.14	72.69	0.05	8.45
$\text{Phi-3}^{\ddagger}_{\text{mini}}$	9.73	5.35	6.10	82.69	1.88	18.76
$\text{Flan-T5}^{\dagger}_{\text{base}}$	29.74	7.67	17.40	82.24	2.30	16.61
$\text{GPT-4o}^{\ddagger}_{\text{mini}}$	43.76	14.33	22.06	84.14	3.96	24.51
PEGASUS	24.32	7.46	15.65	82.74	1.03	10.87
BART	33.53	9.18	18.49	83.28	2.30	16.41
GraphSAGE	38.58	12.13	18.88	83.46	2.80	20.71
GCN	39.31	14.33	20.88	84.04	3.15	22.13
HSG	37.68	14.44	21.88	84.71	5.65	21.36
HAESum	44.67	17.06	23.58	85.31	9.25	28.90
CHANGES	45.89	17.67	22.60	85.20	10.67	31.56
CiteArXiv	46.92	18.63	24.32	**86.12**	11.14	32.87
Ours	**47.25**	**19.57**	**24.76**	85.28	**11.83**	**33.58**

Acronym: R-1 (ROUGE-1), R-2 (ROUGE-2), R-L (ROUGE-L)

∗ Based on extractive ground truth labels

† Extractive-oriented zero-shot prompt, ‡ Abstractive-oriented zero-shot prompt

*The highest result (excluding ORACLE) is **bolded***

4.2 Models for Comparison

We compare our model with four model groups:

- **Unsupervised Models:** ORACLE, which uses extractive ground-truth labels as an approximate upper bound performance (Qi et al., 2022); LEAD, which selects the first few sentences; and two graph-based unsupervised models - LexRank (Erkan and Radev, 2004) and PACSUM (Zheng and Lapata, 2019).

- **General-purpose Language Models:** $\text{Llama}_{\text{tiny}}$ (Touvron et al., 2023), $\text{Phi3}_{\text{mini}}$ (Abdin et al., 2024), $\text{Flan-T5}_{\text{base}}$ (Chung et al., 2024), and $\text{GPT-4o}_{\text{mini}}$ (Hurst et al., 2024) — followed two zero-shot prompting styles: extractive-oriented and abstractive-oriented (Appendix E).

- **Task-specific Language Models:** PEGASUS (Zhang et al., 2020) and BART (Lewis et al., 2020).

- **Deep Neural Graph-based Models:** GraphSAGE (Hamilton et al., 2017), GCN (Yasunaga et al., 2019), HSG (Wang et al., 2020), HAESum (Zhao et al., 2024), CHANGES (Zhang et al., 2023a) and CiteArXiv (Nguyen et al., 2025).

5 Results

We present the main evaluation results of all baseline and proposed models, followed by a performance analysis assessing the contribution of the proposed components.

5.1 Main Result

Table 2 shows the performance of the proposed model compared to related models. Scenario 3 has the lowest score on ORACLE, indicating a lower upper-bound performance compared to others. The LEAD results suggest that key information appears early in Scenarios 1 and 2, unlike Scenario 3, where the abstract sections are not included from the input. Graph-based unsupervised models (LexRank and PACSUM) demonstrate limited performance in most scenarios because they are not trained on labeled data. General-purpose language models were tested with two zero-shot prompting styles: while $\text{Llama}_{\text{tiny}}$ and $\text{Flan-T5}_{\text{base}}$ performed better with the extractive-oriented prompt, $\text{Phi3}_{\text{mini}}$ and $\text{GPT-4o}_{\text{mini}}$ performed better with

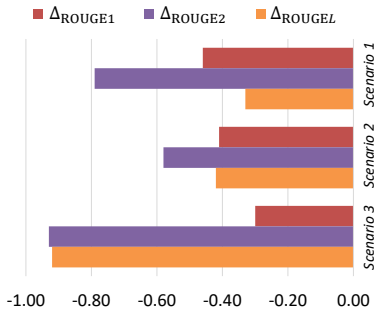

Figure 3: Change in ROUGE scores after excluding citation information from the input. $\Delta_{\text{ROUGE}} = \text{ROUGE}_{without} - \text{ROUGE}_{with}$

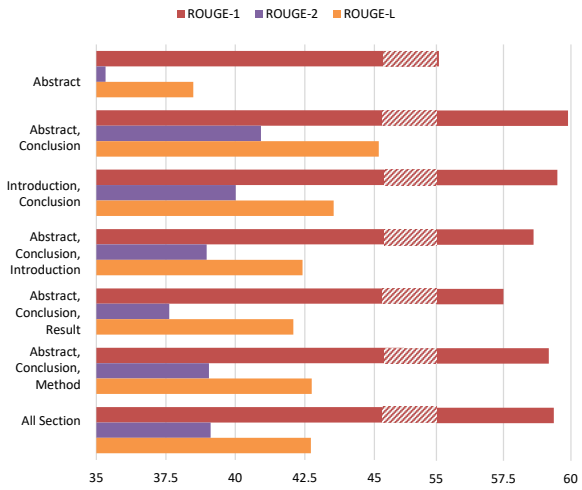

Figure 4: Model performance with different input sections in Scenario 1

the abstractive-oriented prompt (Appendix E). Notably, Flan-T5$_{\text{base}}$ and GPT-4o$_{\text{mini}}$ outperform task-specific language models in most scenarios. In the deep neural graph-based models, CHANGES and CiteArXiv have better results than others.

Across three scenarios, our model consistently outperforms all comparative models. In Scenario 1, our model demonstrates excellent performance across all metrics. This performance surpasses the next best model, CHANGES, which is also based on a heterogeneous graph neural network. Similarly, in Scenario 2, the proposed model maintains its lead, demonstrating its generalization capability in a cross-dataset setting. In Scenario 3, the proposed model continues to have the best performance on a large dataset, demonstrating its generalisation and scalability on the large dataset. The superior performance of our model over other approaches, including those based on heterogeneous graphs (CHANGES and CiteArXiv), stems from its ability to directly incorporate citation information into the document. It also integrates information at varying granularities, enabling context-aware representations and reducing redundancy.

5.2 Performance Analysis

This section demonstrates the effectiveness of incorporating citation information as part of the input to the summarization model. Besides, some experiments are also employed to demonstrate the contributions of the proposed components.

Figure 3 shows the change in ROUGE scores after excluding citations from the input. The negative values of Δ_{ROUGE} scores demonstrate that removing citations from the input degrades model performance across all scenarios and evaluation metrics. The performance drop highlights the crucial role of citations in providing concise, human-written information about the input paper. This is especially valuable when the original document is lengthy and contains complex figures and formulas. Moreover, in Scenario 3, although the summary labels are created automatically from the abstract section, the citation information still contributes significantly to the model's performance.

Some previous studies have shown that important information tends to concentrate in specific positions and sections of scientific documents (Qi et al., 2022). We also evaluate model performance with different input sections, aiming to identify the most informative inputs for our model. However, although scientific documents often follow a common structure, section titles may differ in position and wording. Therefore, to facilitate this experiment, we normalize the section titles by GPT (Appendix F) before feeding them into the model. Figure 4 shows the model performance with different input sections. As a result, the proposed model performs best with input sections: Abstract and Conclusion. Therefore, we select the abstract and conclusion sections as inputs for Scenarios 1 and 2. Since the input documents in Scenario 3 do not include abstracts, the introduction and conclusion sections are chosen.

The architecture of the heterogeneous graph plays a crucial role in determining the model's performance. Figure 5 illustrates the model's performance under various graph configurations, including graphs with a single node type, two node types, three node types, and a fully-connected variant that adds edges between Section and EDU nodes. The results demonstrate that the heteroge-

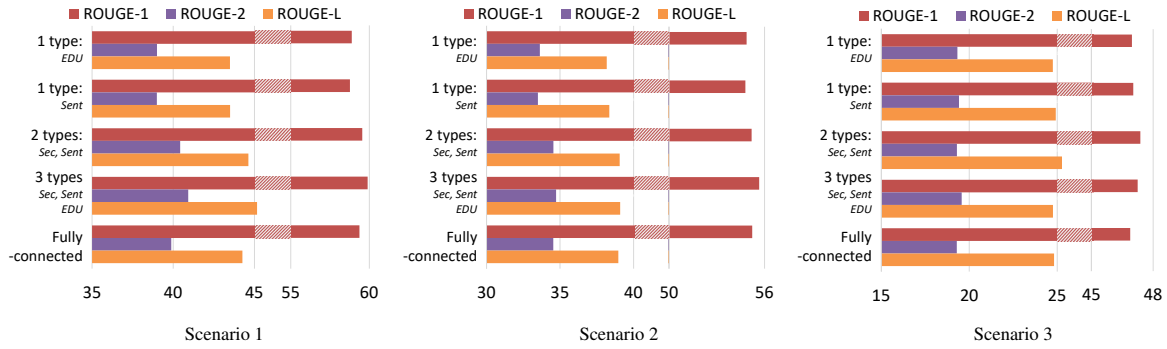

Figure 5: Model performance with different graph structures.

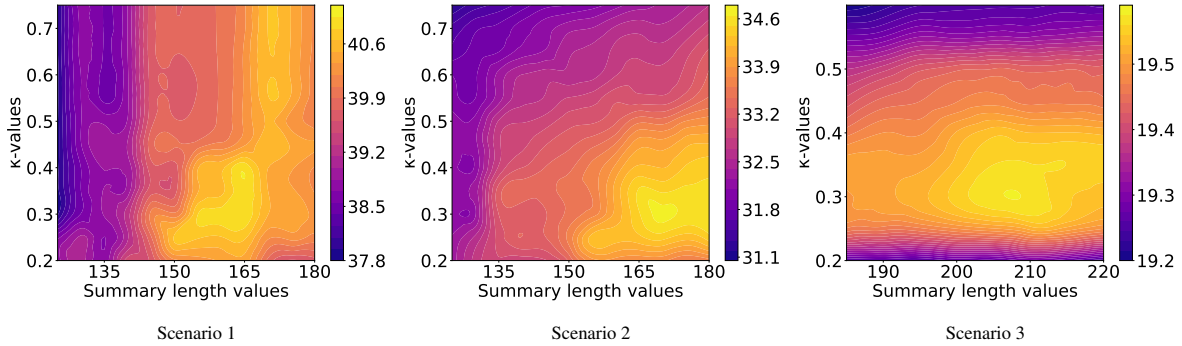

Figure 6: Model performance with different values of maximum summary length and κ.

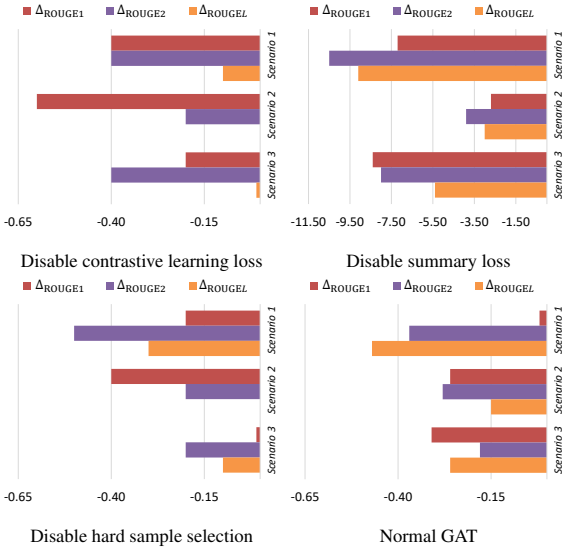

Figure 7: Ablation experiments with components. $\Delta_{\text{ROUGE}} = \text{ROUGE}_{removed} - \text{ROUGE}_{full}$

neous graph with three node types—Section, Sentence, and EDU—achieves the best performance across multiple evaluation metrics. This may be because section nodes help capture structural characteristics, EDU nodes enable the extraction of fine-grained local features and reduce noise, while sentence nodes serve as a bridge between local and global information.

During the inference process, experiments are conducted to determine the optimal values for maximum summary length and the diversity parameter κ across the three scenarios. Figure 6 presents contour plots of ROUGE-2 score with corresponding values of summary lengths and κ. The contour chart shows that a short summary does not provide enough information from the input, while a long summary may include noisy information. The optimal configurations generally match the maximum summary lengths that best reflect the document characteristics of each scenario (165 for Scenario 1, 170 for Scenario 2, and 205 for Scenario 3). Meanwhile, κ values in the range of 0.25 to 0.35, where 0.3 consistently yields strong performance across all scenarios.

Figure 7 shows the ablation and replacement experiments with key components in the proposed model. The experiments demonstrate that all components contribute to the model's performance to different degrees. When disabling the contrastive learning loss, we observe consistent performance degradation across all scenarios, with particularly notable drops in ROUGE-2. Disabling the summary loss also leads to the most severe performance degradation, as it directly optimizes the model based on the predicted labels and the golden labels.

The hard sample selection proves valuable as its removal causes varying performance decreases depending on the scenario, with Scenario 1 showing the largest impact. Finally, the Multi-Granularity Graph Attention is replaced by a normal Graph Attention Network. The performance drop demonstrates that the Multi-Granularity Graph Attention is more effective at learning representations from structured text by employing specialized attention mechanisms for different types of nodes (Section, Sentence, and EDU), thereby capturing both local and global contextual relations.

6 Conclusion

In this paper, we have introduced a heterogeneous graph-based summarization model for scientific documents. Unlike most existing approaches that rely solely on the input document, our model incorporates both the input document and its citations. We proposed a heterogeneous graph architecture that enables the model to capture information across multiple granularities, ranging from fine-grained textual units to the global document structure, and from internal content to external citation context. We also employ a multi-granularity graph encoder to align with the graph architecture and combine triplet loss with summary loss to enhance model performance. Evaluation on three different scenarios shows that our model outperforms related models across all evaluation metrics. Besides, analysis experiments indicate that citation information and proposed components of our model were shown to contribute to its performance.

In the future, we would like to continuously research effective ways to leverage information from scientific documents for summarization.

Limitations

Despite its strong performance, our model has certain limitations that suggest avenues for future improvements. Although incorporating citations shows potential, irrelevant information from citations may affect model performance. Further research is needed to explore noise reduction methods for citations. Additionally, the detailed heterogeneous graph, particularly its fine-grained EDU layer, increases computational resource demands. Optimizing graph construction or exploring pruning techniques to manage this trade-off between detail and resource use needs further research. The lack of rewriting in extractive models leads to inconsistencies in narrative perspective—first-person for text extracted from the document and third-person for text from citations—which may cause confusion for end users.

Acknowledgments

The authors thank the anonymous reviewers for their valuable feedback. Quoc-An Nguyen was funded by the Master, Ph.D. Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2023.ThS.002.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report. *arXiv preprint arXiv:2404.14219*.

Amjad Abu-Jbara and Dragomir Radev. 2011. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 500–509.

Kritika Agrawal, Aakash Mittal, and Vikram Pudi. 2019. Scalable, semi-supervised extraction of structured information from scientific literature. In *Proceedings of the Workshop on Extracting Structured Knowledge from Scientific Publications*, pages 11–20.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: Pretrained language model for scientific text. In *EMNLP*.

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

Lutz Bornmann and Rüdiger Mutz. 2015. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the association for information science and technology*, 66(11):2215–2222.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S Weld. 2020. Tldr: Extreme summarization of scientific documents. *arXiv preprint arXiv:2004.15011*.

Muthu Kumar Chandrasekaran, Guy Feigenblat, Eduard Hovy, Abhilasha Ravichander, Michal Shmueli-Scheuer, and Anita de Waard. 2020. Overview and insights from the shared tasks at scholarly document processing 2020: Cl-scisumm, laysumm and longsumm. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 214–224.

Janara Christensen and 1 others. 2013. Towards coherent multi-document summarization. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1163–1173.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert systems with applications*, 165:113679.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *NeurIPS*, 30.

Yin Jou Huang and Sadao Kurohashi. 2021. Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3046–3052.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 7871. Association for Computational Linguistics.

Zhenwen Li, Wenhao Wu, and Sujian Li. 2020. Composing elementary discourse units in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196.

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

Zhengyuan Liu, Ke Shi, and Nancy Chen. 2021. Dmrst: A joint framework for document-level multilingual rst discourse segmentation and parsing. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 154–164.

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.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishnan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 584–592.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31-1.

Quoc-An Nguyen, Xuan-Hung Le, Thi-Minh-Thu Vu, and Hoang-Quynh Le. 2025. Citearxiv: A citation-enriched dataset and heterogeneous graph-based model for scientific articles summarization. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 3084–3087.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11670–11688.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Vahed Qazvinian, Dragomir Radev, and Arzucan Özgür. 2010. Citation summarization through keyphrase extraction. In *Proceedings of the 23rd international conference on computational linguistics (COLING 2010)*, pages 895–903.

Vahed Qazvinian and Dragomir R Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics-COLING'08*. Association for Computational Linguistics.

Siya Qi, Lei Li, Yiyang Li, Jin Jiang, Dingxin Hu, Yuze Li, Yingqi Zhu, Yanquan Zhou, Marina Litvak, and Natalia Vanetik. 2022. Sapgraph: Structure-aware extractive summarization for scientific papers with heterogeneous graph. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association*

for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 575–586.

Shahbaz Syed, Ahmad Hakimi, Khalid Al Khatib, and Martin Potthast. 2023. Citance-contextualized summarization of scientific papers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8551–8568.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuan-Jing 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.

Jie Wang, Shutian Ma, and Chengzhi Zhang. 2017. Citationas: A summary generation tool based on clustering of retrieved citation content. *Framework*, 7(8):19–27.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7386–7393. AAAI Press.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Contrastive hierarchical discourse graph for scientific document summarization. In *4th Workshop on Computational Approaches to Discourse*, page 37.

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

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pages 11328–11339. PMLR.

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

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. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663.

A EDU Parsing

In this study, sentence segmentation into EDUs is performed using a joint framework for document-level RST discourse segmentation and parsing (Liu et al., 2021). This model, named DMRST, is designed to handle multilingual discourse parsing and provides a mechanism for identifying the boundaries of EDUs within a given text.

In detail, an input text is first processed by the framework through a shared encoder to generate contextualized representations for the entire text. Subsequently, the two tasks are trained jointly, allowing parsing information to guide the segmentation process and vice versa, thereby enhancing the accuracy of both. The details of the two tasks are as follows:

1. **Segmentation:** EDU segmentation is formulated as a sequence labeling task, where the model predicts boundary labels for tokens to identify the limits of discourse units.

2. **Parsing:** A hierarchical RST discourse tree is constructed to represent the rhetorical organization of the text, specifying the relationships (e.g., Elaboration, Contrast) that hold between the segmented EDUs.

The final output is a complete discourse tree. The leaf nodes of this tree correspond to the identified EDUs. We use the pre-trained model provided by the authors, which is referenced in our released source code, to ensure consistency and reproducibility.

B Hard Sample Selection

Selecting hard samples has proven to enhance the model's performance (Ostendorff et al., 2022). Figure 8 shows a hard sample example in vector space.

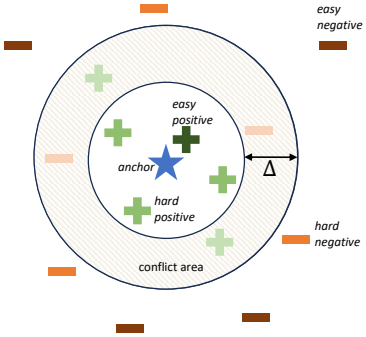

Figure 8: Hard samples in vector space

Hard negatives are close to but do not overlap with positives, challenging the model to distinguish differences. Meanwhile, hard positives are the samples that are close but not trivially close to the anchor vector, helping the model learn diversity. To avoid positive and negative samples colliding (ensuring that samples are clearly positive or negative), we selected a margin Δ to separate these two types of samples. This margin (described in **Formula 10**) allows the use of threshold values to exclude uncertain cases.

Secondly, the model selects hard positives as the top k EDUs with the lowest scores from the positive samples and hard negatives as the top k EDUs with the highest scores from the negative samples.

C Hyperparameter Configuration

Table 3: Hyperparameter settings across scenarios

Hyperparameter	Scenario 1 High-quality	Scenario 2 Cross	Scenario 3 Large
Hidden size	128	128	256
Heads	64	64	64
Dropout	0.3	0.3	0.3
Learning rate	0.0003	0.0003	0.0003
Negative threshold (γ)	0.4	0.4	0.4
Positive threshold (β)	0.5	0.5	0.5
Triplet loss margin (α)	2	2	2
Triplet hard samples (k)	3	3	5
Loss scaling factor (λ)	0.5	0.5	0.5
Maximum summary length	165	170	205
Diversity controller (κ)	0.3	0.3	0.3

Acronym: sim (similarity)

Table 3 presents the hyperparameter settings used across the three experimental scenarios. The model architecture maintains consistent configurations across scenarios, with hidden sizes of 128 for Scenarios 1 and 2, increasing to 256 for the more complex Scenario 3. All scenarios use 64 attention heads and a dropout rate of 0.3 to prevent overfitting. Training settings remain consistent with a learning rate of 0.0003. For triplet loss, we set positive/negative threshold to $0.4/0.5$ and a triplet loss margin α of 2, with the number of hard samples k adjusted from 3 (Scenarios 1 and 2) to 5 (Scenario 3). The training-loss scaling factor is tuned in the range $[0, 1]$ with a step size of 0.1, and 0.5 is found to offer the best balance between convergence speed and accuracy. In inference, the maximum summary length is set to 165 words for Scenario 1, 170 for Scenario 2, and 205 for the more complex Scenario 3, with a diversity controller κ of 0.3 across all scenarios.

D Computational Cost Analysis

To provide a clear overview of the computational overhead, we present a detailed comparison in Table 4. The results demonstrate that while the introduction of the EDU-level graph increases the number of parameters, the impact on training and inference time is modest, ensuring the model remains practical and scalable.

Table 4: Comparison of Computational Cost and Model Parameters.

Metric	Without EDU	With EDU
Parameters	2.71M	3.30M
Training Time *(50 epochs, 40 samples)*	~242s	~303s
Inference Time *(11852 tokens)*	0.092s	0.109s

E LM Prompting Strategy

With general-purpose language models, we adopted the zero-shot prompting strategies for extractive and abstractive summarization (Zhang et al., 2023b). This approach uses a two-part structure—a system role definition and a user command—to guide the model's behavior.

Extractive-oriented prompt The objective is to require the model to select important sentences from the source text. The model is assigned the role of an extractive summarizer and instructed to choose the top-k important sentences, ensuring that

Table 5: Comparing Extractive-oriented and Abstractive-oriented Zero-Shot Prompts for Summarization on General-Purpose Language Models

Model	Extractive-oriented						Abstractive-oriented					
	R-1	R-2	R-L	BERT	BLEU	Meteor	R-1	R-2	R-L	BERT	BLEU	Meteor
Scenario 1: High-quality Dataset												
$Llama_{tiny}$	**10.16**	**7.01**	**8.11**	**84.97**	**3.02**	**18.79**	8.82	6.42	7.19	85.35	2.77	17.39
$Phi\text{-}3_{mini}$	10.37	7.12	8.15	84.98	3.07	18.64	**10.42**	**7.17**	**8.21**	**85.50**	**3.12**	**18.70**
$Flan\text{-}T5_{base}$	**51.05**	**31.17**	**36.79**	**87.75**	**19.33**	**34.69**	48.00	29.19	35.43	87.33	16.64	31.83
$GPT\text{-}4o_{mini}$	46.15	18.89	26.38	86.93	8.00	35.17	**50.81**	**31.03**	**34.49**	**87.93**	**20.99**	**42.53**
Scenario 2: Cross-Dataset												
$Llama_{tiny}$	**10.06**	**7.06**	**8.04**	**84.75**	**3.03**	**18.84**	9.22	6.35	7.41	84.64	2.69	17.33
$Phi\text{-}3_{mini}$	10.20	7.15	8.09	84.76	3.07	18.72	**10.24**	**7.19**	**8.13**	**85.22**	**3.10**	**18.82**
$Flan\text{-}T5_{base}$	**48.21**	**27.56**	**34.45**	**87.40**	**16.13**	**30.52**	46.82	26.39	32.90	86.96	15.10	30.70
$GPT\text{-}4o_{mini}$	45.59	18.05	25.86	86.78	7.62	34.57	**50.30**	**28.61**	**33.06**	**87.57**	**19.12**	**41.20**
Scenario 3: Large Dataset												
$Llama_{tiny}$	**14.81**	**1.24**	**11.14**	**72.69**	**0.05**	**8.45**	14.31	1.11	10.35	73.31	0.02	8.01
$Phi\text{-}3_{mini}$	9.72	5.32	6.08	82.45	1.87	18.78	**9.73**	**5.35**	**6.10**	**82.69**	**1.88**	**18.76**
$Flan\text{-}T5_{base}$	**29.74**	**7.67**	**17.40**	**82.24**	**2.30**	**16.61**	26.94	7.14	17.03	81.59	1.79	14.21
$GPT\text{-}4o_{mini}$	40.29	13.52	19.93	83.59	3.46	27.77	**43.76**	**14.33**	**22.06**	**84.14**	**3.96**	**24.51**

Acronym: R-1 (ROUGE-1), R-2 (ROUGE-2), R-L (ROUGE-L)

*The higher result for each model corresponding to a prompt style is highlighted in **bolded***

Table 6: Section Title Normalization Prompt

Your task is to extract the section names from the following scientific article and classify each section into one of the following labels: [Abstract, Introduction, Method, Result, Conclusion, Others].
Return a response where the section name and its corresponding label are printed on a single line in the format: (section's name — label).

the final output is a subset of the original document and follows the length constraint.

Abstractive Summarization Prompt The objective is to encourage the model to generate a new summary by synthesizing information. In this prompt, the model is assigned the role of an abstractive summarizer, which allows it to freely rephrase, reorganize, and condense the source content.

The detailed results of the models using this prompt are reported in Table 5. While $Llama_{tiny}$ and $Flan\text{-}T5_{base}$ performed better with the extractive-oriented prompt, $Phi3_{mini}$ and $GPT\text{-}4o_{mini}$ performed better with the abstractive-oriented prompt.

F Section Title Normalization

Although scientific documents generally follow a common structure (abstract, introduction, related Work, method, results, and conclusion), section titles may differ in positions and wording. To address this challenge, we leverage GPT to classify section titles into six main categories: *Abstract*, *Introduction*, *Method*, *Result*, *Conclusion*, and *Others*. Table 6 presents the prompt used to guide the process.

In addition, we observed that different wordings are often used to refer to the same type of section. For example, *summary*, *conclusion*, *conclusions*, and *conclusion & future work* all typically indicate the Conclusion section. To ensure consistency, we created a mapping dictionary to normalize such variations into unified labels.