Hierarchical Attention Graph for Scientific Document Summarization in Global and Local Level

Chenlong Zhao[1,2] , Xiwen Zhou[1,2] , Xiaopeng Xie[1,2], Yong Zhang[1,2]*
[1]School of Electronic Engineering,
Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Beijing Key Laboratory of Work Safety Intelligent Monitoring,
Beijing University of Posts and Telecommunications, Beijing 100876, China
{Chenlong_000325,zhouxiwen,657314qq,yongzhang}@bupt.edu.cn

Abstract

Scientific document summarization has been a challenging task due to the long structure of the input text. The long input hinders the simultaneous effective modeling of both global high-order relations between sentences and local intra-sentence relations which is the most critical step in extractive summarization. However, existing methods mostly focus on one type of relation, neglecting the simultaneous effective modeling of both relations, which can lead to insufficient learning of semantic representations. In this paper, we propose HAESum, a novel approach utilizing graph neural networks to locally and globally model documents based on their hierarchical discourse structure. First, intra-sentence relations are learned using a local heterogeneous graph. Subsequently, a novel hypergraph self-attention layer is introduced to further enhance the characterization of high-order inter-sentence relations. We validate our approach on two benchmark datasets, and the experimental results demonstrate the effectiveness of HAESum and the importance of considering hierarchical structures in modeling long scientific documents[1].

1 Introduction

Extractive summarization aims to select a set of sentences from the input document that best represents the information of the whole document. With the advancement of pre-trained models and neural networks over the years, researchers have achieved promising results in news summarization (Liu and Lapata, 2019; Zhong et al., 2020). However, when applying these methods to long scientific documents, they encounter challenges due to the relatively lengthy inputs. The considerable length of the text hinders sequential models from capturing both long-range dependencies across sentences and intra-sentence relations simultaneously (Wang et al., 2020).Moreover, the extended context exceeds the input limits of the Transformer-based model (Vaswani et al., 2017) due to the quadratic computational complexity of self-attention.

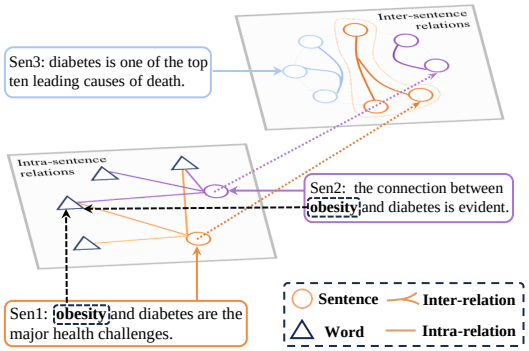

Figure 1: An illustration of modeling an input document from local and global perspectives. Triangles and circles represent words and sentences in the original document respectively.

Recently, the application of large language models (LLM) such as ChatGPT to text summarization tasks has gained significant interest and attracted widespread attention. A recent study by (Zhang et al., 2023b) evaluated the performance of ChatGPT on extractive summarization and further enhanced its performance through in-context learning and chain-of-thought. Another study (Ravaut et al., 2023) conducted experiments on abstractive summarization using various LLMs on a variety of datasets that included long inputs. While the use of LLMs in text summarization tasks has demonstrated exciting potential, there are still several limitations that have not been addressed. The most important of these is the phenomenon of lost-in-the-middle (Liu et al., 2023; Ravaut et al., 2023), where LLMs ignore information in the middle and pay more attention to the context at the beginning and end. This bias raises concerns especially in sum-

*Corresponding author
[1]Our code will be available at https://github.com/MoLICHENXI/HAESum

Findings of the Association for Computational Linguistics: NAACL 2024, pages 714–726
June 16-21, 2024

marization tasks where important text may be scattered throughout the document (Wu et al., 2023). Additionally, as the input length increases, even on explicitly long-context models, the model's performance gradually declines (Liu et al., 2023).

As a result, researchers have turned to graph neural networks to model long-distance relations. They represent a document as a graph and update node representations in the graph using message passing. These works use different methods to construct a graph from documents, such as using sentence similarity as edge weights to model cross-sentence relations (Zheng and Lapata, 2019). Another popular approach is to construct a word-document heterogeneous graph (Wang et al., 2020), using words as intermediate connecting sentences. Phan et al. (2022) further added passage nodes to the heterogeneous graph to enhance the semantic information. Zhang et al. (2022) proposed a hypergraph transformer to capture high-order cross-sentence relations.

Despite the impressive success of these approaches, we observe that the current work still lacks a comprehensive consideration on relational modeling. More specifically, two limitations are mentioned: (1) Most of the existing approaches focus on modeling intra-sentence relations but often overlook cross-sentence high-order relations. Inter-sentence connections may not only be pairwise but could also involve triplets or higher-order relations (Ding et al., 2020). In the hierarchical discourse structure of scientific documents, sentences within the same section often express the same main idea. It is difficult to fully understand the content of a document by merely considering intra-sentence and cross-sentence relations in pairwise. (2) These approaches rely on updating relations at different levels simultaneously but ignore the hierarchical structure of scientific documents. Sentences are composed of words and, in turn, contribute to forming sections. By understanding the meaning of individual tokens, we get the meaning of the sentence and thus the content of the section. Therefore, bottom-to-top structured modeling is crucial to understand the content of the document.

To address the above challenges, we propose HAESum (**H**ierarchical **A**ttention Graph for **E**xtractive Document **Sum**marization), a method that leverages a graph neural network model to fully explore hierarchical structural information in scientific documents. HAESum first constructs a local heterogeneous graph of word-sentence and updates sentence representations at the intra-sentence level. The local sentence representations are then fed into a novel hypergraph self-attention layer to further update and learn the cross-sentence sentence representations through a self-attention mechanism that fully captures the relations between nodes and edges. Figure 1 is an illustration showing the modeling of local and global context information from a hierarchical point of view, and the resulting representations contain both local and global hierarchical information. We validate HAESum with extensive experiments on two benchmark datasets and the experimental results demonstrate the effectiveness of our proposed method. In particular, we highlight our main contributions as follows:

(i) We introduce a novel graph-based model utilizing the hierarchical structure of scientific documents for modeling. In contrast to simultaneously updating nodes in the graph, we learn intra-sentence and inter-sentence relations separately from both local and global perspectives. To the best of our knowledge, we are the first approach to hierarchical modeling using different graphs on this task.

(ii) We propose a novel hypergraph self-attention layer that utilizes the self-attention mechanism to further aggregate high-order sentence representations. Moreover, our approach does not rely on pre-trained models as encoders, making it easily applicable to other low-resource languages.

(iii) We validate our model on two benchmark datasets, and the experimental results demonstrate the effectiveness of our approach against strong baselines.

2 Related Work

2.1 Scientific Paper Summarization

Scientific document summarization has been a hot topic due to the challenges of modeling long texts (Frermann and Klementiev, 2019). Cohan et al. (2018) introduced two benchmark datasets for long documents, Arxiv and PubMed, and employed a hierarchical encoder and discourse-aware decoder for the document summarization task. Cui and Hu (2021) proposed a sliding selector network accompanied by dynamic memory to alleviate information loss between context segments. Gu et al. (2021) presented a reinforcement learning-based method that achieved impressive performance by considering the extraction history at each time step. Recently, Ruan et al. (2022) proposed a method to inject explicit hierarchical structural information

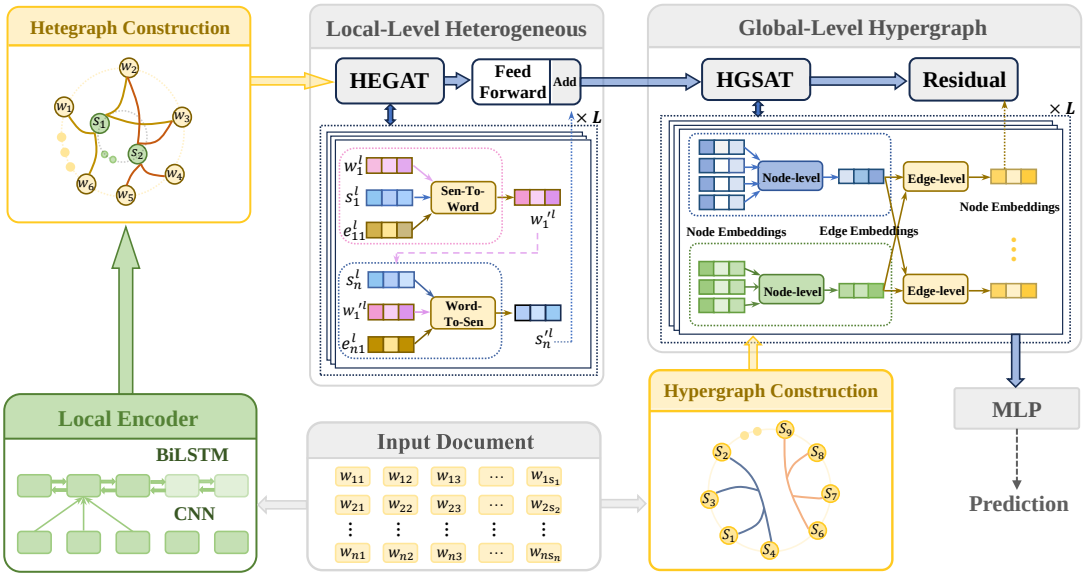

Figure 2: Overview of the proposed HAESum framework. We first build a local-level heterogeneous graph (HEGAT) for the input document and apply message passing to iteratively update the representation in two stages: sentence-to-word and word-to-sentence. The obtained sentence representations are then fed into the hypergraph self-attention layer (HGSAT) to obtain the global representations and used for the final sentence selection.

such as section titles and sentence positions into a pre-trained model to further improve the performance and interpretability.

2.2 Graph based Summarization

Graph neural networks have been widely used for extractive summarization due to their flexibility and scalability. Dong et al. (2020) proposed an unsupervised graph-based model that combines both sentence similarity and hierarchical discourse structure to rank sentences. Cui et al. (2020) injected latent topic information into graph neural networks to further improve performance. Wang et al. (2020) constructed a word-document heterogeneous graph using word nodes as intermediate to connect sentences. Zhang et al. (2022) proposed a hypergraph transformer to model long-distance dependency while emphasizing the importance of high-order inter-sentence relations in extraction summarization. Our paper follows this line of work, but the main difference is that our approach combines both intra-sentence relations and high-order cross-sentence relations and efficiently leverages the hierarchical discourse structure of scientific documents to learn sentence representations that incorporate both local and global information.

3 Method

Given an arbitrary document $D = \{s_1, s_2, ..., s_n\}$ consisting of n sentences, each sentence consists of m words $s_i = \{w_{i1}, w_{i2}, ..., w_{im}\}$. The goal of extractive summarization is to predict labels $y_i \in \{0, 1\}$ for all sentences, where $y_i = 1$ indicates that the current sentence should be included in the summary. The overall structure of HAESum is shown in Figure 2.

3.1 Local-level Heterogeneous Graph

As the lowest level of the hierarchical structure, in this section, we will first introduce how to capture local intra-sentence relations between sentences and their corresponding words using a heterogeneous graph. We will start by explaining how to construct the heterogeneous graph and initialize it, followed by detailing how to use a heterogeneous self-attention layer to update node representations. Finally, we will feed the updated sentence node representations into the next module.

3.1.1 Graph Construction

Given an input document D, we first construct a heterogeneous graph $G = \{V, E\}$, where V represents a set of nodes and E represents edges between nodes. In order to utilize the natural hierarchy between words and sentences of a document, the nodes can be defined as $V = V_w \cup V_s$, where $V_w = \{w_1, w_2..., w_n\}$ denotes n different words in the document, and $V_s = \{s_1, s_2, ..., s_m\}$ denotes the m sentences in the document. The edges are defined as $E = \{e_{11}, e_{12}, ..., e_{mn}\}$, where e_{ij} is

a real-valued edge weight that denotes the cross-connection between a sentence node i and a word node j contained by it.

3.1.2 Graph Initializers

Let $X_w \in R^{|V_w| \times d_w}$, $X_s \in R^{|V_s| \times d_s}$ denote the feature matrices of the input word and sentence respectively. d_w and d_s correspond to the feature dimensions of words and sentences, respectively. We first use Glove (Pennington et al., 2014) to initialize word representations. Instead of using pre-trained model as a sentence encoder, we first use CNN (LeCun et al., 1998) with different kernel sizes to get the n-gram feature S^C of the sentence followed by using BiLSTM (Hochreiter and Schmidhuber, 1997) to obtain the sentence-level feature S^B.The features obtained from CNN and BiLSTM are concatenated as initialized sentence representations $X_S = \text{Cat}(S^C, S^B)$.

3.1.3 Heterogeneous Attention Modules

Following the previous work (Wang et al., 2020), we employ the heterogeneous graph attention layer for node representations updating. Specifically, when a node v_i aggregates information from its neighbours, the attention coefficient α_{ij} for node v_j is computed as follows:

$$z_{ij} = \text{LeakyReLU}(W_a[W_s h_i \| W_k h_j]; e_{ij}) \quad (1)$$

$$\alpha_{ij} = \frac{\exp(z_{ij})}{\sum_{l \in \mathcal{N}} \exp(z_{il})} \quad (2)$$

where W_a, W_s, W_k are trainable weights. $\|$ denotes concatenation. We also inject the edge features e_{ij} into the attention mechanism for computation.

We also add multi-head attention and Feed-Forward layer (FFN) (Vaswani et al., 2017) to further improve the performance. The final representation $u_i^{'}$ of node v_i is then obtained as follows:

$$u_i = \|_{k=1}^{K} \sigma(\sum_{j \in \mathcal{N}} \alpha_{ij}^k W^k h_i) \quad (3)$$

$$u_l^{'} = \text{FFN}(u_i) + h_i \quad (4)$$

We begin by aggregating the sentence nodes around the word to update word representations. Subsequently, we utilize the updated word representations to further update the sentence representations.

In this section, we use the local heterogeneous graph to learn the intra-sentence relations at the lowest level of the document hierarchy.

3.2 Global-level Hypergraph

In this section, we first introduce how to construct a hypergraph. Subsequently, we present a novel hypergraph self-attention layer designed to fully capture high-order global inter-sentence relations. Finally, the resulting sentence representations are used to decide whether to include them in the summary.

3.2.1 Hypergraph Construction

A hypergraph is defined as $G = \{V, E\}$, where $V = \{v_1, v_2, ..., v_n\}$ represents a set of nodes and $E = \{e_1, e_2..., e_n\}$ represents hyperedges in the graph. Unlike edges in regular graphs, hyperedges can connect two or more nodes and thus represent multivariate relations. A hypergraph is typically represented by its incidence matrix $\mathbf{H} \in R^{n \times m}$:

$$\mathbf{H}_{ij} = \begin{cases} 1, \text{if } v_i \in e_j \\ 0, \text{if } v_i \notin e_j \end{cases} \quad (5)$$

where $v_i \in V$, $e_j \in E$ and if the hyperedge e_j connects node v_i there is $v_i \in e_j$.

We denote a sentence s_i in a document $D = \{s_1, s_2, ..., s_n\}$ as a node v_i in the hypergraph. In order to capture global higher-order inter-sentence relations, we consider creating section hyperedges for each part (Suppe, 1998). A hyperedge e_j will be created if a set of child nodes $V_j \in V$ belongs to the same section in the document. The node representations in the hypergraph are initialized to the output of the previous module.

The initialized node features $H_{sen} = \{h_1, h_2, ..., h_n\} \in R^{n \times d}$ and incidence matrix H will be fed into the hypergraph self-attention network to learn effective sentence representations.

3.2.2 Hypergraph Self-Attention Modules

Hypergraph attention networks (HGAT) are designed to learn node representations using a mutual attention mechanism. This mutual attention mechanism divides the computational process into two steps, i.e., node aggregation and hyperedge aggregation. First the hyperedge representations are updated with node information. Subsequently, the hyperedge information is fused back to the nodes from hyperedges.

The HGAT has mainly been implemented based on graph attention mechanism (Veličković et al., 2017), such as HyperGAT (Ding et al., 2020). However, this attention mechanism employs the same weight matrix for different types of nodes and hyperedges information and could not fully exploit

the relations between nodes and hyperedges, which prevents the model from capturing higher-order cross-sentence relations (Fan et al., 2021).

To address the limitations of HGAT, we propose the hypergraph self-attention layer. Inspired by the success of Transformer (Vaswani et al., 2017) in textual representation and graph learning (Ying et al., 2021), we use the self-attention mechanism to fully explore the relations between nodes and hyperedges. The entire structure we propose is described below.

Node-level Attention To solve the problem of initializing the hyperedge features, we first encode hyperedge representations from node aggregation information using node-level attention. Given node features $H_{sen}^{l-1} = \{h_1^{l-1}, h_2^{l-1}, ..., h_n^{l-1}\}$ and incidence matrix, hyperedge representations $\{f_1^l, f_2^l..., f_m^l\}$ can be computed as follows:

$$f_j^l = \text{LeakyReLU}(\sum_{s_k \in e_j} \alpha_{jk} W_n h_k^{l-1}) \quad (6)$$

$$\alpha_{jk} = \frac{\exp(W_h^{\mathrm{T}} u_k)}{\sum_{s_l \in e_j} \exp(W_h^{\mathrm{T}} u_l)} \quad (7)$$

$$u_k = \text{LeakyReLU}(W_p h_k^{l-1}) \quad (8)$$

where the superscript l denotes the model layer. W_n, W_h, W_p are trainable parameters. α_{jk} is the attention coefficient of node s_k in the hyperedge e_j. Through the node-level attention mechanism, we initialize the hyperedge representation.

Edge-level Attention As an inverse procedure, the self-attention mechanism is applied to compute the importance scores to highlight the hyperedges that are more critical for the next layer of node representation v_i. Given the node feature matrix H_{sen}^{l-1} and the hyperedge feature matrix F_{edge}^l, similar to the self-attention mechanism we compute the output matrix as follows:

$$\begin{aligned} Q_{sen}^{l-1} &= W_q H_{sen}^{l-1} \\ K_{edge}^{l} &= W_k F_{edge}^{l} \\ V_{edge}^{l} &= W_v F_{edge}^{l} \end{aligned} \quad (9)$$

$$\text{Att}(H, F) = \text{softmax}(\frac{Q_{sen}^{l-1} {K_{edge}^{l}}^{\mathrm{T}}}{\sqrt{d_k}}) V_{edge}^{l} \quad (10)$$

where W_q,W_k,W_v are trainable parameters. d_k is the feature dimension of the hidden layer. Att() represents the self-attention mechanism.

After obtaining the enhanced node representations H_{sen}^l using the hypergraph self-attention layer, we applied a feature fusion layer to generate the final representations $H_{sen}^{'l}$, which can be represented by the formula:

$$H_{sen}^{'l} = \text{LeakyReLU}(W_1 H_{sen}^{l-1} \| W_2 H_{sen}^{l}) \quad (11)$$

$\|$ denotes concatenation. Fusing hyperedge information and node information, we obtain a semantic representation of sentence nodes.

Datasets	Document			Avg.	Avg.
	Train	Val	Test	Doc.	Token.
Arxiv	202703	6436	6439	4938	220
PubMed	116669	6630	6657	3016	203

Table 1: Statistics of Arxiv and PubMed datasets.

3.3 Opimization

After passing L hypergraph self-attention layers, we obtain the representations of sentences $H_{sen} = \{h_1, h_2, ..., h_n\} \in R^{n \times d}$. We then add a multi-layer perceptron (MLP) followed by a LayerNorm layer and obtain a score $\hat{y}_i$, indicating whether it will be selected as a summary. Formally, the prediction score for a sentence node s_i is computed as follows:

$$\hat{y}_i = W_o(\text{LayerNorm}(W_p h_i)) \quad (12)$$

where W_o,W_p are trainable parameters.

Finally, the output sentence scores $\hat{y}_i$ are optimized with the true labels y_i by binary cross-entropy loss:

$$L = \frac{1}{N} \sum_{i=1}^{N} y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \quad (13)$$

where N denotes the number of sentences in the document.

4 Experiment

4.1 Experiment setup

We validate our proposed model on two scientific document datasets and compare it to the strong baselines. In the following, we start with the details of the datasets.

Datasets We perform extensive experiments on two benchmark datasets: Arxiv and PubMed (Cohan et al., 2018). Arxiv is a long document dataset containing different scientific domains. PubMed contains articles in the biomedical domain. We use the original train, validation, and testing splits as in

Models	PubMed			Arxiv		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Oracle	55.05	27.48	49.11	53.88	23.05	46.54
PacSum	39.79	14.00	36.09	38.57	10.93	34.33
HIPORANK	43.58	17.00	39.31	39.34	12.56	34.89
FAR	41.98	15.66	37.58	40.92	13.75	35.56
ExtSum-LG	44.85	19.70	31.43	43.62	17.36	29.14
Topic-GraphSum	45.95	20.81	33.97	44.03	18.52	32.41
SSN-DM	46.73	21.00	34.10	45.03	19.03	32.58
HEGEL	47.13	21.00	42.18	46.41	18.17	39.89
MTGNN	48.42	22.26	43.66	46.39	18.58	40.50
HiStruct+	46.59	20.39	42.11	45.22	17.67	40.16
CHANGES	46.43	21.17	41.58	45.61	18.02	40.06
TLM-I+E	42.13	16.27	39.21	41.62	14.69	38.03
PEGASUS	45.49	19.90	42.42	44.70	17.27	25.80
BigBird	46.32	20.65	42.33	46.63	19.02	**41.77**
Dancer	46.34	19.97	42.42	45.01	17.60	40.56
ChatGLM3-6B-32k	40.95	15.79	37.09	39.81	14.14	35.36
HAESum (ours)	**48.77**	**22.44**	**43.83**	**47.24**	**19.44**	41.34

Table 2: Experimental Results on PubMed and Arxiv datasets. We report ROUGE scores from the original papers if available, or scores from (Xiao and Carenini, 2019) otherwise.

(Cohan et al., 2018). Detailed statistics for the two benchmark datasets are shown in Table 1.

Compared Baselines We make a systematic comparison with recent approaches in this area. We categorize these methods into the following four types:

- Unsupervised methods: graph-based models PacSum (Zheng and Lapata, 2019), HIPORANK (Dong et al., 2020), FAR (Liang et al., 2021).
- Neural extractive model: Seq2Seq-based models HiStruct+ (Ruan et al., 2022); local and global context model ExtSum-LG (Xiao and Carenini, 2019); graph-based models Topic-GraphSum (Cui et al., 2020), SSN-DM (Cui and Hu, 2021), HEGEL (Zhang et al., 2022), MTGNN (Doan et al., 2022), CHANGES (Zhang et al., 2023a).
- Neural abstractive model: encoder-decoder based Model TLM-I+E (Pilault et al., 2020), PEGASUS (Zhang et al., 2020) , BigBird (Zaheer et al., 2020), divide-and-conquer approach Dancer (Gidiotis and Tsoumakas, 2020).
- Large language model: ChatGLM3-6k-32k (Zeng et al., 2022). More details on the evaluation of the large language model can be found in Appendix A.1.

4.2 Implementation Details

Regarding the encoding of word nodes, the vocabulary size is 50000 and the word embedding is initialized with a dimension of 300 using the Glove pretrained model(Pennington et al., 2014). The feature dimensions of sentence nodes and edges in the heterogeneous graph are set to 64 and 50, respectively. The hyperedge feature dimension is 64. We set the maximum sentence length of each document to 200 and the maximum number of words per sentence to 100. In our experiments, we stacked two layers of heterogeneous graph attention modules (HEGAT) and hypergraph self-attention modules (HSAGT). The multi-head of the HEGAT layer is set to 8 and 6, respectively.

The model is optimized using the Adam optimizer (Loshchilov and Hutter, 2017) with a learning rate of 0.0001 and a dropout rate of 0.1. We train the model on an RTX A6000 GPU with 48GB of memory for 12 epochs. The training process stops if the validation set loss does not decrease three times. The training time for one epoch on the PubMed dataset is 3 hours, while on the Arxiv dataset, it is 6 hours.

We use a greedy search algorithm similar to (Nallapati et al., 2017) to select sentences from documents as the gold extractive summaries (Oracle). Following previous work, we use ROUGE (Lin and Hovy, 2003) to evaluate the quality of summaries.

Model	ROUGE-1	ROUGE-2	ROUGE-L
PubMed			
HAESum	**48.77**	**22.44**	**43.83**
w/o Heterogeneous	47.45	21.12	42.56
w/o HyperAttention	47.60	21.43	42.78
Arxiv			
HAESum	**47.24**	**19.44**	**41.34**
w/o Heterogeneous	46.91	19.22	41.03
w/o HyperAttention	46.75	19.01	40.91

Table 3: Ablation study results on PubMed and Arxiv datasets.

We use ROUGE-1/2 to measure summary informativeness and ROUGE-L to measure the fluency of the summary.

4.3 Experiment Results

Table 2 shows the comparison between our model HAESum and the baseline model on PubMed and Arxiv datasets. The first block covers the ground truth ORACLE and unsupervised methods for extractive summarization. The second block covers state-of-the-art supervised extractive baselines. The third block reports abstractive methods.

Based on the results, we find that HIPORANK (Dong et al., 2020) achieves strong performance on graph-based unsupervised modeling. Compared to other unsupervised methods, HIPORANK adds section information, which demonstrates the effectiveness and importance of taking the natural hierarchical structure of scientific documents into account when modeling cross-sentence relations.

In the extractive baseline, MTGNN (Doan et al., 2022) achieves state-of-art performance, MTGNN considers more intra-sentence level modeling, which shows the necessity of modeling from low-level structure. HEGEL (Zhang et al., 2022) is the most similar approach to ours. HEGEL injects external information such as keywords and topics into the model and models higher-order cross-sentence relations through a hypergraph transformer to achieve a competitive performance. However, compared to MTGNN, HEGEL does not consider low-level intra-sentence relations, which proves the necessity of considering and modeling hierarchical structure. Interestingly, CHANGES (Zhang et al., 2023a) achieves equally impressive results in hierarchical modeling by considering high-level intra-section and inter-section relations, further confirming the importance of hierarchical modeling. Among the extractive methods, the transformer-based HiStruct+ (Ruan et al., 2022) shows a competitive performance, which demonstrates the effectiveness of the self-attention mechanism. HiStruct+ also incorporates the inherent hierarchical structure into the pre-trained language models to achieve strong performance. In addition, the extractive approaches largely outperform the abstractive approaches, which may be due to the fact that long input is more challenging for the decoding process of the abstractive models.

Method	ROUGE-1	ROUGE-2	ROUGE-L
Hierarchical(Ours)	**48.77**	**22.44**	**43.83**
Parallelization	48.36	22.03	43.36

Table 4: Different ways of updating sentence representations on PubMed dataset.

Through the table, the results of using the large language model are not satisfactory compared to our proposed method. By analyzing the output of the large language model, the model sometimes incorrectly outputs content from other languages and also occasionally outputs duplicate content. In addition, the model sometimes misinterprets extractive summarization as abstractive summarization.

According to the experimental results, our model HAESum outperforms all extractive and abstractive strong baselines. In particular, our model neither requires injection of external knowledge (e.g., topics and keywords (Zhang et al., 2022)) to enhance global information nor pre-trained model's (e.g., BERT (Devlin et al., 2018)) knowledge (Doan et al., 2022). The outstanding performance of HAESum demonstrates the importance of hierarchical modeling of local intra-sentence relations and global inter-sentence relations.

5 Analysis

5.1 Ablation Study

We first analyze the effect of different components of HAESum in Table 3. The second row shows that removing the heterogeneous graph part represents not learning intra-sentence relations. The third row removes the hypergraph component, representing the absence of learning higher-order cross-sentence relations. As shown in table 3, removing either part hurts the model performance, which indicates that learning both local intra-sentence relations and global higher-order cross-sentence relations is necessary for scientific document summarization.

Interestingly, these two components are almost equally important for modeling long documents. This indicates the importance of simultaneously

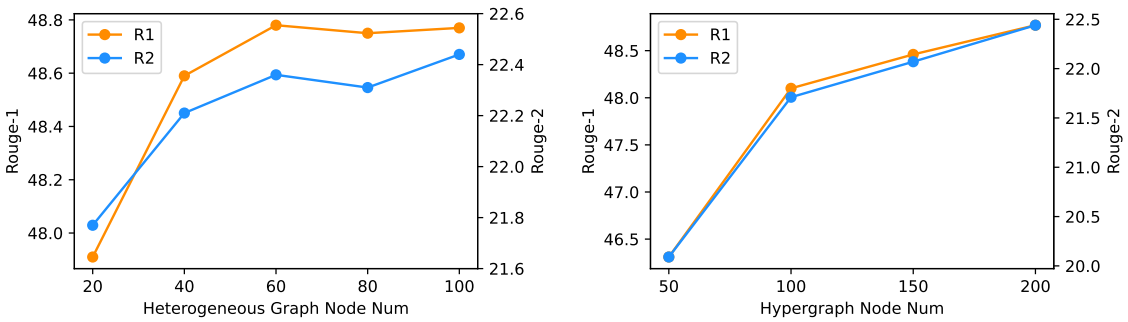

Figure 3: ROUGE-1,2 performance of HAESum with different number of graph nodes on PubMed dataset.

Model	ROUGE-1	ROUGE-2	ROUGE-L
PubMed			
with HSGAT (ours)	**48.77**	**22.44**	**43.83**
with HGAT	48.64	22.25	43.64
Arxiv			
with HSGAT (ours)	**47.24**	**19.44**	**41.34**
with HGAT	47.08	19.26	41.18

Table 5: Different attention mechanism results on PubMed and Arxiv datasets.

modeling semantic aspects from diverse perspectives and hierarchical discourse structures in scientific documents.

5.2 Performance Analysis

Hierarchical discourse We also analyze different update approaches for obtaining the final sentence representations in HAESum. As shown in Table 4, the second row represents our hierarchical updating. The third row represents parallel updating, where intra-sentence and inter-sentence relations are updated simultaneously, and the final sentence representations are concatenated. The superior performance of hierarchical updating over parallel updating once again emphasizes the critical importance of the bottom-to-top modeling sequence we propose for understanding the content of long documents.

Attention mechanism We then analyze the performance of our proposed novel hypergraph self-attention layer and hypergraph attention network (HGAT). As shown in Table 5, our hypergraph self-attention layer outperforms HGAT (Ding et al., 2020). We speculate that the main reason is the utilization of the self-attention mechanism and different weight matrices, which fully exploit relations between nodes and edges, thereby enhancing the learning of high-order relations.

Hyperparameter sensitivity In our experiments, we set the maximum input length for each sentence to be 100, and the maximum sentence length for each input document to be 200. We conduct an analysis of these two hyperparameters. In addition, more information about the distribution of the sentence lengths and the number of sentences in the document is presented in the Appendix A.3. As shown in Figure 3, when the maximum number of tokens in each sentence is reduced from 100 to 60, the performance does not significantly decrease. This indicates that under this range of hyperparameter settings, the model has already processed most of the tokens in each sentence. However, as the length continues to decrease, the model's performance starts to decline, as the input length limits the capture of local intra-sentence relations.

Simultaneously, when the maximum number of sentences in a document is increased from 50 to 200, the model's performance continues to improve. This improvement is attributed to the consideration of more sentences, capturing more complex higher-order cross-sentence relations. However, persistently increasing this hyperparameter leads to significant computational consumption. Specifically, in future work, we intend to increase the maximum input sentences per document while minimizing computational consumption as much as possible.

5.3 Case Study

Here we provide an example of a summary output by HAESum, as shown in Table 6. The selected sentences are mainly from the same section and cover the entire document. This illustrates that HAESum can effectively learn both local intra-sentence and high-order inter-sentence relations, facilitating the selection of the most relevant sentences.

(Introduction) It includes hidradenitis suppurativa acne conglobata dissecting cellulitis of the scalp and pilonidal sinus. **(Introduction)** Though each of these conditions are commonly encountered on their own as a symptom complex follicular occlusion tetrad has rarely been reported in the literature here. **(Introduction)** We present a case of hidradenitis suppurativa in a 36-year-old male patient who also had the above mentioned associations. **(Case Report)** A 36-year-old male patient presented to us with a history of recurrent boils since 18 years. **(Discussion)** Follicular occlusion tetrad is a condition that includes hidradenitis suppurativa (hs) acne conglobata dissecting cellulitis of the scalp and pilonidal sinus.

Table 6: An example output summary of our proposed model.

6 Conclusion

This paper presents HAESum for scientific document summarization. HAESum employs a graph-based model to comprehensively learn local intra-sentence and high-order inter-sentence relations, utilizing the hierarchical discourse structure of scientific documents for modeling. The impressive performance of HAESum demonstrates the importance of simultaneously considering multiple perspectives of semantics and hierarchical structural information in modeling scientific documents.

Limitations

Despite the outstanding performance of our HAESum, several limitations are acknowledged. Firstly, HAESum solely leverages intra-sentence and inter-sentence relations in scientific documents. We believe that incorporating other hierarchical discourse structures at different granularities, such as sentence-section information (Zhang et al., 2023a) or dependency parsing trees, could further enhance model performance. Secondly, although the context window sizes of large language models satisfy the input length of scientific documents, their performance on text summarization tasks, especially on long input texts, remains to be improved due to the loss-in-the-middle (Liu et al., 2023; Ravaut et al., 2023) problem. We consider this issue as a future work. Additionally, we focused on single document summarization. We believe that incorporating domain knowledge through citation networks and similar methods could further improve performance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant (No.61971057).

References

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. *arXiv preprint arXiv:1804.05685*.

Peng Cui and Le Hu. 2021. Sliding selector network with dynamic memory for extractive summarization of long documents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5881–5891.

Peng Cui, Le Hu, and Yuanchao Liu. 2020. Enhancing extractive text summarization with topic-aware graph neural networks. *arXiv preprint arXiv:2010.06253*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. *arXiv preprint arXiv:2011.00387*.

Xuan-Dung Doan, Minh Le Nguyen, and Khac-Hoai Nam Bui. 2022. Multi graph neural network for extractive long document summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5870–5875.

Yue Dong, Andrei Mircea, and Jackie CK Cheung. 2020. Discourse-aware unsupervised summarization of long scientific documents. *arXiv preprint arXiv:2005.00513*.

Haoyi Fan, Fengbin Zhang, Yuxuan Wei, Zuoyong Li, Changqing Zou, Yue Gao, and Qionghai Dai. 2021. Heterogeneous hypergraph variational autoencoder for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4125–4138.

Lea Frermann and Alexandre Klementiev. 2019. Inducing document structure for aspect-based summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6263–6273.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040.

Nianlong Gu, Elliott Ash, and Richard HR Hahnloser. 2021. Memsum: Extractive summarization of long documents using multi-step episodic markov decision processes. *arXiv preprint arXiv:2107.08929*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Improving unsupervised extractive summarization with facet-aware modeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1685–1697.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics*, pages 150–157.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ling Luo, Xiang Ao, Yan Song, Feiyang Pan, Min Yang, and Qing He. 2019. Reading like her: Human reading inspired extractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3033–3043.

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.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Tuan Anh Phan, Ngoc-Dung Ngoc Nguyen, and Khac-Hoai Nam Bui. 2022. Hetergraphlongsum: heterogeneous graph neural network with passage aggregation for extractive long document summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6248–6258.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319.

Mathieu Ravaut, Shafiq Joty, Aixin Sun, and Nancy F Chen. 2023. On position bias in summarization with large language models. *arXiv preprint arXiv:2310.10570*.

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

Frederick Suppe. 1998. The structure of a scientific paper. *Philosophy of Science*, 65(3):381–405.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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.

Yunshu Wu, Hayate Iso, Pouya Pezeshkpour, Nikita Bhutani, and Estevam Hruschka. 2023. Less is more for long document summary evaluation by llms. *arXiv preprint arXiv:2309.07382*.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. *arXiv preprint arXiv:1909.08089*.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2022. Hegel: Hypergraph transformer for long document summarization. *arXiv preprint arXiv:2210.04126*.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Contrastive hierarchical discourse graph for scientific document summarization. *arXiv preprint arXiv:2306.00177*.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023b. Extractive summarization via chatgpt for faithful summary generation. *arXiv preprint arXiv:2304.04193*.

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.

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

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. *arXiv preprint arXiv:2004.08795*.

A Appendix

In this section, we give more details about the experiment.

A.1 Evaluation on LLMs

We tested different prompts and chose the best prompt. The **prompt** we used is: You are given a long scientific literature. Please read and choose no more than five sentences from the original scientific literature as a summary. Scientific literature:[*Text Document*]. Now, select no more than five sentences from the original given scientific literature as a summary. Summary:[*Output*].

The experimental results are shown in Table 7, where we considered a variety of possible large language models. However, in order to fulfill the requirement of inputting long texts, we chose ChatGLM3-6B-32K (Zeng et al., 2022) to evaluate the performance results on two datasets.

Through the table, the results of using the large language model are not satisfactory compared to our proposed method. By analyzing the output of the large language model, the model sometimes incorrectly outputs content from other languages and also occasionally outputs duplicate content. In addition, the model sometimes misinterprets extractive summarization as abstractive summarization. The most serious problem is that the model still pays too much attention to the context **at the beginning and end**. Our approach takes into account both intra-sentence and inter-sentence relationships, and effectively extracts key sentences distributed throughout the context and uses them as summaries. In addition, our model satisfies the input length constraints and saves computational resources.

A.2 Human Evaluation

We conduct human evaluation following the previous work (Luo et al., 2019). We randomly sample 50 documents from the test sets of PubMed and Arxiv and ask three volunteers to evaluate the summaries extracted by HAESum, MTGNN, and LLM. For each document-summary pair, they are asked to rank them on three aspects: overall quality, coverage and non-redundancy. Notably the best one will be marked rank 1 and so on, and if both models extracted the same summaries they will both be ranked the same. We report the average results over the two datasets in Table 8

As seen through the table, our method achieves better results compared to other baselines. The human evaluation also further validates the effectiveness of our proposed method.

A.3 Distribution of Sentence Length and Number of Tokens in the Dataset

In order to better demonstrate the validity of our choice of hyperparameters, we counted the distribution of sentence lengths in PubMed dataset as well as the distribution of the number of sentences. The experimental results are shown in Table 9

The obtained table shows that the hyperparameters we chose cover almost all the range of the distribution. This is further evidence that the choice of hyperparameters in the *Hyperparameter sensitivity* section is adequate and effective.

Models	Satisfy The Input Length	PubMed			Arxiv		
		ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
ChatGPT	✘	-	-	-	-	-	-
LLaMa-7B	✘	-	-	-	-	-	-
ChatGLM3-6B	✘	-	-	-	-	-	-
ChatGLM3-6B-32k	✔	40.95	15.79	37.09	39.81	14.14	35.36
HAESum(ours)	✔	**48.77**	**22.44**	**43.83**	**47.24**	**19.44**	**41.34**

Table 7: Experimental results on large language models on two datasets

Model	PubMed			Arxiv		
	Overall	Coverage	Non-Redundancy	Overall	Coverage	Non-Redundancy
ChatGLM3-6B-32K	2.52	2.51	2.41	2.48	2.45	2.29
MTGNN	1.73	1.74	**1.67**	1.85	1.91	1.83
HAESum(Ours)	**1.68**	**1.64**	1.71	**1.61**	**1.57**	**1.68**

Table 8: Average rank of human evaluation in terms of overall performance, coverage, and non-redundancy. Lower score is better.

The distribution of the sentence lengths					
(0, 20]	(20, 40]	(40, 60]	(60, 80]	(80, 100]	Over 100
29.63%	51.08%	12.82%	3.78%	1.38%	1.31%
The distribution of the number of sentences					
(0, 50]	(50, 100]	(100, 150]	(150, 200]	(200, 250]	Over 250
28.09%	40.83%	19.36%	7.59%	2.57%	1.56%

Table 9: The distribution of the sentence lengths and the number of sentences in PubMed dataset