

Contrastive Hierarchical Discourse Graph for Scientific Document Summarization

Haopeng Zhang, Xiao Liu, Jiawei Zhang

IFM Lab, Department of Computer Science, University of California, Davis, CA, USA

haopeng,xiao,jiawei@ifmlab.org

Abstract

The extended structural context has made scientific paper summarization a challenging task. This paper proposes CHANGES, a contrastive hierarchical graph neural network for extractive scientific paper summarization. CHANGES represents a scientific paper with a hierarchical discourse graph and learns effective sentence representations with dedicated designed hierarchical graph information aggregation. We also propose a graph contrastive learning module to learn global theme-aware sentence representations. Extensive experiments on the PubMed and arXiv benchmark datasets prove the effectiveness of CHANGES and the importance of capturing hierarchical structure information in modeling scientific papers.

1 Introduction

Extractive document summarization aims to extract the most salient sentences from the original document and form the summary as an aggregate of these sentences. Compared to abstractive summarization approaches that suffer from hallucination generation problems (Kryściński et al., 2019; Zhang et al., 2022b), summaries generated in an extractive manner are more fluent, faithful, and grammatically accurate, but may lack coherence across sentences. Recent advances in deep neural networks and pre-trained language models (Devlin et al., 2018; Lewis et al., 2019) have led to significant progress in single document summarization (Nallapati et al., 2016a; Narayan et al., 2018; Liu and Lapata, 2019; Zhong et al., 2020). However, these methods mainly focus on short documents like news articles in CNN/DailyMail (Hermann et al., 2015) and New York Times (Sandhaus, 2008), and struggle when dealing with relatively long documents such as scientific papers.

The challenges of lengthy scientific paper summarization lie in several aspects. First, the extended input context hinders cross-sentence relation modeling, the critical step of extractive summarization

(Wang et al., 2020). Thus, sequential models like RNN are incapable of capturing the long-distance dependency between sentences, and hard to differentiate salient sentences from others. Furthermore, scientific papers tend to cover diverse topics and contain rich hierarchical discourse structure information. The internal hierarchy structure, like sections, paragraphs, sentences, and words, is too complex for sequential models to capture. Scientific papers generally follow a standard discourse structure of problem definition, methodology, experiments and analysis, and conclusions (Xiao and Carenini, 2019). Moreover, the lengthy input context also makes the widely adopted self-attention Transformer-based models (Vaswani et al., 2017) inapplicable. The input length of a scientific paper can range from 2000 to 7,000 words, which exceeds the input limit of the Transformer due to the quadratic computation complexity of self-attention. Thus, sparse Transformer models like BigBird (Zaheer et al., 2020) and Longformer (Beltagy et al., 2020) are proposed.

Recently, researchers have also turned to graph neural networks (GNN) as an alternative approach. Graph neural networks have been demonstrated to be effective at tasks with rich relational structure and can preserve global structure information (Yao et al., 2019; Xu et al., 2019; Zhang and Zhang, 2020). By representing a document as a graph, GNNs update and learn sentence representations by message passing, and turn extractive summarization into a node classification problem. Among all attempts, one popular way is to construct cross-sentence similarity graphs (Erkan and Radev, 2004; Zheng and Lapata, 2019), which uses sentence representation cosine similarity as edge weights to model cross-sentence semantic relations. Xu et al. (2019) proposed using Rhetorical Structure Theory (RST) trees and coreference mentions to capture cross-sentence discourse relations. Wang et al. (2020) proposed constructing a word-document het-

erogeneous graph by using words as the intermediary between sentences. Despite their success, how to construct an effective graph to capture the hierarchical structure for academic papers remains an open question.

To address the above challenges, we propose CHANGES (Contrastive Hierarchical Graph neural network for Extractive Summarization), a hierarchical graph neural network model to fully exploit the section structure of scientific papers. CHANGES first constructs a sentence-section hierarchical graph for a scientific paper, and then learns *hierarchical* sentence representations by dedicated designed information aggregation with iterative intra-section and inter-section message passing. Inspired by recent advances in contrastive learning (Liu and Liu, 2021; Chen et al., 2020), we also propose a graph contrastive learning module to learn global theme-aware sentence representations and provide fine-grained discriminative information. The local sentence and global section representations are then fused for salient sentence prediction. We validate CHANGES with extensive experiments and analyses on two scientific paper summarization datasets. Experimental results demonstrate the effectiveness of our proposed method. Our main contributions are as follows:

- We propose a hierarchical graph-based model for long scientific paper extractive summarization. Our method utilizes the hierarchical discourse of scientific documents and learns effective sentence representations with iterative intra-section and inter-section sentence message passing.
- We propose a plug-and-play graph contrastive module to provide fine-grained discriminative information. The graph contrastive module learns global theme-aware sentence representations by pulling semantically salient neighbors together and pushing apart unimportant sentences. Note that the module could be added to any extractive summarization system.
- We validate our proposed model on two benchmark datasets (arXiv and PubMed), and the experimental results demonstrate its effectiveness over strong baselines.

2 Related Work

2.1 Extractive Summarization on Scientific Papers

Despite the superior performance on news summarization by recent neural network models (Zhou et al., 2018; Zhang et al., 2023a,b; Fonseca et al., 2022) and pre-trained language models (Liu and Lapata, 2019; Lewis et al., 2019), progress in long document summarization such as scientific papers is still limited.

Traditional approaches to summarize scientific articles rely on supervised machine learning algorithms such as LSTM (Collins et al., 2017) with surface features such as sentence position, and section categories. Recently, Xiao and Carenini (2019) proposed a neural-based method by incorporating both the global context of the whole document and the local context within the current topic with an encoder-decoder model. Ju et al. (2021) designed an unsupervised extractive approach to summarize long scientific documents based on the Information Bottleneck principle. Dong et al. (2020) proposed an unsupervised ranking model by incorporating two-level hierarchical graph representation and asymmetrical positional cues to determine sentence importance. Recent works also apply pre-trained sparse language models like Longformer for modeling long documents (Beltagy et al., 2020; Ruan et al., 2022; Cho et al., 2022).

2.2 Graph-based Summarization

Graph models have been widely applied to extractive summarization due to the capability of modeling cross-sentence relations within a document. The sparsity nature of graph structure also brings scalability and flexibility, making it a good fit for long documents. Graph neural networks’ memory costs are generally linear with regard to the input size compared to the quadratic self-attention mechanism.

Researchers have explored supervised graph neural network methods for summarization (Cui and Hu, 2021; Jia et al., 2020; Huang and Kurohashi, 2021; Xie et al., 2022; Phan et al., 2022). Yasunaga et al. (2017) first proposed to use Graph Convolutional Network (GCN) on the approximate discourse graph. Xu et al. (2019) then applied GCN on structural discourse graphs based on RST trees and coreference mentions. Recently, Wang et al. (2020) proposed constructing a word-document heterogeneous graph by using words as the intermediary

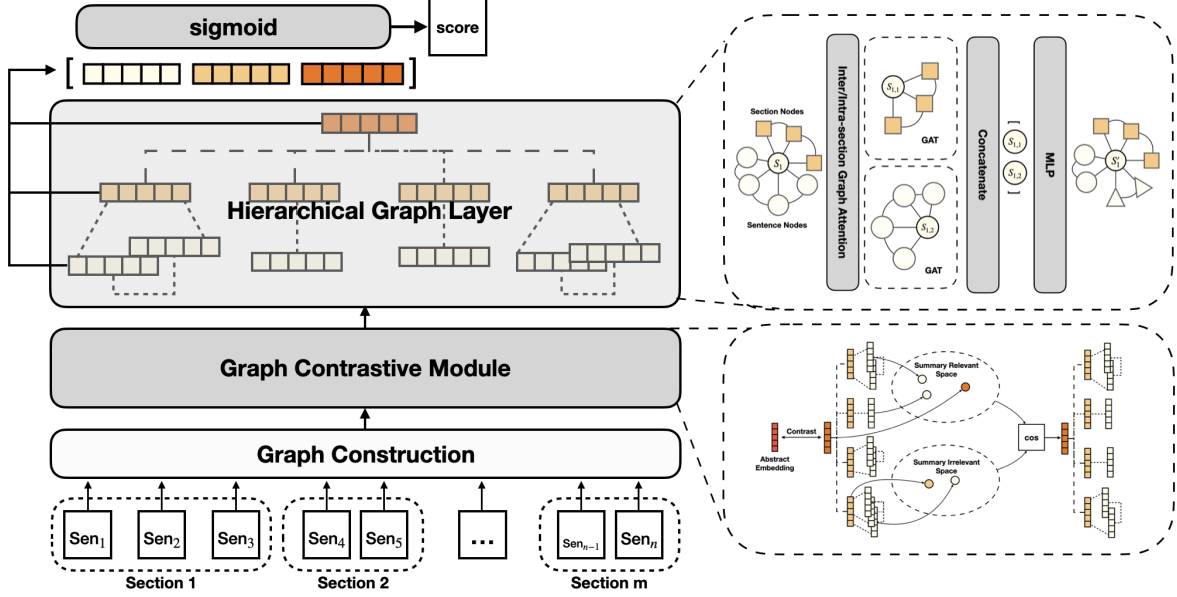


Figure 1: The overall model architecture of CHANGES. We first construct a hierarchical graph for an input document, and then learn representations with a graph contrastive module and hierarchical graph layers. The concatenation representations of sentence node and its section node will be fused for summary sentence selection.

between sentences. Zhang et al. (2022a) proposed to use hypergraph to capture the high-order sentence relations within the document. Our paper follows the series of work but incorporates hierarchical graphs for scientific paper discourse structure modeling and graph contrastive learning for theme-aware sentence representation learning.

3 Method

Given a document $D = \{s_1, s_2, \dots, s_n\}$ with n sentences and m sections, we first represent it as a hierarchical graph and formulate extractive summarization as a node labeling task. The objective is to predict labels $y_i \in (0, 1)$ for all sentences, where $y_i = 1$ and $y_i = 0$ represent whether the i -th sentence should be included in the summary or not, respectively.

The overall model architecture of CHANGES is shown in Figure 1. CHANGES consists of two modules: a graph contrastive learning module to learn global theme-aware sentence representations and a hierarchical graph layer module to learn hierarchical graph node representations with iterative message passing. The learned sentence node and section node representations will be used as indicators for salient sentence selection.

3.1 Graph Construction

Given an academic paper D , we first construct a hierarchical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} stands for the node set and \mathcal{E} represents edges between nodes. In order to utilize the sentence-section hierarchical structure of academic papers, the undirected hierarchical graph \mathcal{G} contains both sentence nodes and section nodes, defined by $\mathcal{V} = \mathcal{V}_{sen} \cup \mathcal{V}_{sec}$, where each sentence node $v_{sen_i} \in \mathcal{V}_{sen}$ represents a sentence s_i in the document D and $v_{sec_j} \in \mathcal{V}_{sec}$ represents one section in the document. The edge connection of \mathcal{G} is defined as $\mathcal{E} = \mathcal{E}_{sen} \cup \mathcal{E}_{sec} \cup \mathcal{E}_{cross}$, where \mathcal{E}_{sen} denotes the connection between sentence nodes within the same section, \mathcal{E}_{sec} denotes the connection between section nodes, and \mathcal{E}_{cross} denotes the cross-connection between a sentence node and its corresponding section node. Note that we also add a special section supernode v_D that represents the whole document D . An illustration of the hierarchical graph is shown in Figure 2.

Edge Connection Unlike prior work (Zheng and Lapata, 2019; Dong et al., 2020) that uses cosine similarity of sentence semantic representations as edge weights, we construct unweighted hierarchical graphs to disentangle structural information (adjacency matrix \mathbf{A}) from semantic information (node representation \mathbf{H}). In other words, connected nodes have weight 1, and disconnected nodes have

weight 0 in the adjacency matrix \mathbf{A} .

Formally, sentence-level edge $e_{sen_{i,j}}$ connects sentence nodes v_{sen_i} and v_{sen_j} if they are within the same section, aiming to aggregate local intra-section information. All section nodes are fully connected by section-level edges $e_{sec_{p,q}}$, aiming to aggregate global inter-section information. The cross-level edge $e_{cross_{i,p}}$ connects the sentence node v_{sen_i} to its corresponding section node v_{sec_p} , which allow message passing between sentence-level and section-level nodes.

In a hierarchical graph, a sentence node could only directly interact with local neighbor nodes within the same section, and indirectly interact with sentence nodes of other sections via section-level node connections.

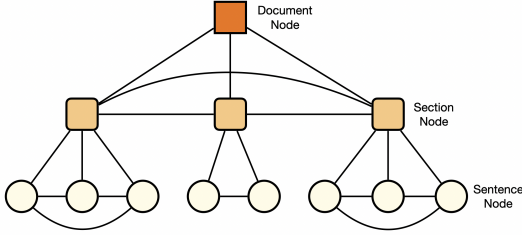


Figure 2: An illustration of a hierarchical graph for a long input document with rich discourse structures.

Node Representation We adopt BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) as sentence encoder to embed the semantic meanings of sentences $\{s_1, s_2, \dots, s_n\}$ as initial node representations $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$. Note that BERT here is only used for initial sentence embedding, but is not updated during the training process to reduce model computing cost and increase efficiency.

In addition to the semantic representation of sentences, we also inject positional encoding following Transformer (Vaswani et al., 2017) to preserve the sequential order information. We apply the hierarchical position embedding by (Ruan et al., 2022) to model sentence positions accompanying our hierarchical graph. Specifically, the position of each sentence s_i can be represented as two parts: its corresponding section index p_i^{sec} , and its sentence index within section p_i^{sen} . Formally, the hierarchical position embedding (HPE) of sentence s_i can be calculated as:

$$\text{HPE}(s_i) = \text{PE}(p_i^{sec}) + \text{PE}(p_i^{sen}), \quad (1)$$

where $\text{PE}(\cdot)$ refers to the position encoding func-

tion in (Vaswani et al., 2017):

$$\text{PE}(\text{pos}, 2i) = \sin(\text{pos}/10000^{2i/d}), \quad (2)$$

$$\text{PE}(\text{pos}, 2i + 1) = \cos(\text{pos}/10000^{2i/d}). \quad (3)$$

Overall, we can get the initial sentence node representations $\mathbf{H}_{sen}^0 = \{\mathbf{h}_{sen_1}^0, \mathbf{h}_{sen_2}^0, \dots, \mathbf{h}_{sen_n}^0\}$, with vector $\mathbf{h}_i^0 \in \mathbb{R}^d$ defined as:

$$\mathbf{h}_i^0 = \mathbf{x}_i + \text{HPE}(s_i), \quad (4)$$

where d is the dimension of the node embedding. The initial section node representation $\mathbf{h}_{sec_j}^0 \in \mathbb{R}^d$ for the j -th section is the mean of its connected sentences embeddings, and the document node representation $\mathbf{h}_{doc}^0 \in \mathbb{R}^d$ is the mean of all section node embeddings.

3.2 Graph Contrastive Module

After constructing the hierarchical graph with adjacency matrix \mathbf{A} and node representation $\mathbf{H}_{sen}^0 \in \mathbb{R}^{n \times d}$, we apply a graph contrastive learning (GCL) module to capture global context information. Motivated by the principal idea that a good summary sentence should be more semantically similar to the source document than the unqualified sentences (Radev et al., 2004; Zhong et al., 2020), our GCL module updates sentence representations using Graph Attention Network (Veličković et al., 2017) with a contrastive objective to learn the global theme-aware sentence representations. Note that the module could be added to any extractive summarization system.

Graph Attention Network Given a constructed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node representations \mathbf{H} and adjacent matrix \mathbf{A} , a GAT layer updates a node v_i with representation \mathbf{h}_i by:

$$\begin{aligned} e_{ij} &= \text{LeakyReLU}(\mathbf{W}_a [\mathbf{W}_{in} \mathbf{h}_i \parallel \mathbf{W}_{in} \mathbf{h}_j]), \\ \alpha_{ij} &= \frac{\exp(e_{ij})}{\sum_{l \in \mathcal{N}_i} \exp(e_{il})}, \\ \mathbf{h}'_i &= \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}_v \mathbf{h}_j \right), \end{aligned} \quad (5)$$

where \mathcal{N}_i denotes the 1-hop neighbors of node v_i , α_{ij} denotes the attention weight between nodes \mathbf{h}_i and \mathbf{h}_j , \mathbf{W}_{in} , \mathbf{W}_a , \mathbf{W}_v are trainable weight matrices, and \parallel denotes concatenation operation.

The above single-head graph attention is further extended to multi-head attention, where T independent attention mechanisms are conducted and their outputs are concatenated as:

$$\mathbf{h}'_i = \parallel_{t=1}^T \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^t \mathbf{W}_h^k \mathbf{h}_j \right) \quad (6)$$

Contrastive Loss Contrastive learning aims to learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors (Marelli et al., 2014). Recent works have demonstrated contrastive learning to be effective in high-order representation learning (Chen et al., 2020; Gao et al., 2021). Thus, we optimize our GCL module in a contrastive manner with the following contrastive loss. The goal of contrastive learning is to learn theme-aware sentence embedding by pulling semantically salient neighbors together and pushing apart less salient sentences. The contrastive loss is formally defined as:

$$\mathcal{L}_c = -\log \frac{\exp(\text{sim}((\mathbf{h}'_D, \mathbf{h}'_i)/\tau))}{\sum_{j=1}^n \exp(\text{sim}(\mathbf{h}'_D, \mathbf{h}'_j)/\tau)}, \quad (7)$$

where \mathbf{h}'_D is the document node embedding, \mathbf{h}'_i is the updated representation of sentence s_i , and τ is the temperature factor.

After passing through the GCL module, the learned global theme-aware sentence embeddings $\mathbf{H}_{sen}^c = \{\mathbf{h}_{sen_1}^c, \mathbf{h}_{sen_2}^c, \dots, \mathbf{h}_{sen_n}^c\} \in \mathbb{R}^{n \times d}$ are then passed to the hierarchical graph layer module.

3.3 Hierarchical Graph Layer

To exploit the sentence structure of academic papers, CHANGES then updates sentence and section node representations with hierarchical graph layers in an iterative manner.

The hierarchical graph layer first updates sentence embeddings with the local neighbor sentences within the same section with GAT for intra-section message passing, then update section nodes with sentence nodes for cross-level information aggregation to exploit the hierarchical structure of academic papers. Next, inter-section message passing allow *global* context information interaction. Finally, the sentence nodes are updated based on their corresponding section node, fusing both local and global context information.

Formally, each iteration contains four update processes: one intra-section message passing,

	Arxiv	PubMed
# train	201,427	112,291
# validation	6,431	6,402
# test	6,436	6,449
avg. word/doc	4,938	3,016
avg. word/summary	203	220
avg. sent./doc	205	140
avg. sent./summary	5	6

Table 1: Statistics of PubMed and Arxiv datasets.

one sentence-to-section aggregation, one inter-section message passing, and finally one section-to-sentence aggregation. For the l -th iteration, the process can be represented as:

$$\begin{aligned} \mathbf{H}'_{sen} &= \text{GAT}(\mathbf{H}_{sen}^l) \\ \mathbf{H}'_{sec} &= \text{GAT}(\mathbf{H}'_{sen}) \\ \mathbf{H}_{sec}^{l+1} &= \text{GAT}(\mathbf{H}'_{sec}) \\ \mathbf{H}_{sen}^{l+1} &= \sigma(\mathbf{W}_b[\mathbf{H}'_{sen} \parallel \mathbf{H}_{sec}^{l+1}]) \end{aligned} \quad (8)$$

where $\mathbf{H}'_{sen}, \mathbf{H}'_{sec}$ denotes the intermediate representations of sentence and section nodes, $\mathbf{H}_{sen}^{l+1}, \mathbf{H}_{sec}^{l+1}$ denotes the updated sentence and section node representations, and $[\mathbf{H}'_{sen} \parallel \mathbf{H}_{sec}^{l+1}]$ denotes the concatenation of intermediate sentence node representation and its corresponding updated section node representation.

In this way, CHANGES updates and learns hierarchy-aware sentence embeddings through the hierarchical graph layers.

3.4 Optimization

After passing L hierarchical graph layers, we obtain the final sentence node representations $\mathbf{H}_{sen}^L = \{\mathbf{h}_{sen_1}^L, \mathbf{h}_{sen_2}^L, \dots, \mathbf{h}_{sen_n}^L\} \in \mathbb{R}^{n \times d}$. We then add a multi-layer perceptron (MLP) followed by a sigmoid activation function to indicate the confidence scores for extracting each sentence in the summary.

Formally, the predicted confidence score \hat{y}_i to extract a sentence s_i in section sec_j as a summary sentence is:

$$\mathbf{z}_i = \text{LeakyReLU}(\mathbf{W}_{o1}[\mathbf{h}_{sen_i}^L \parallel \mathbf{h}_{sec_j}^L]), \quad (9)$$

$$\hat{y}_i = \text{sigmoid}(\mathbf{W}_{o2}\mathbf{z}_i), \quad (10)$$

where $\mathbf{W}_{o1}, \mathbf{W}_{o2}$ are both trainable parameters, and $[\mathbf{h}_{sen_i}^L \parallel \mathbf{h}_{sec_j}^L]$ denote the concatenation of sentence embedding and its corresponding section embedding. During the inference phase, we will select the k sentences with the highest predicted confidence scores as the extractive summary for the input long document.

Since the extractive ground truth labels for long documents are highly imbalanced, we optimize hierarchical graph layers using weighted cross entropy loss following (Xiao and Carenini, 2019) as:

$$\mathcal{L}_s = -\frac{1}{NN_d} \sum_{d=1}^N \sum_{i=1}^{N_d} (\eta \cdot y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)), \quad (11)$$

where N denotes the number of training instances in the training set, N_d denotes the number of sentences in the document, $\eta = \frac{\#negative}{\#positive}$ denote the ratio of the number of negative and positive sentences in the document, and y_i represent the ground-truth of sentence i .

Training Loss Overall, we optimize CHANGES in an end-to-end manner, by optimizing the graph contrastive module and hierarchical graph layers simultaneously.

The overall training loss of CHANGES is:

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_c, \quad (12)$$

where λ is a re-scale hyperparameter and \mathcal{L}_c denotes the contrastive loss in Equation 7.

4 Experiment

4.1 Experiment Setup

Dataset To validate the effectiveness of CHANGES, we conduct extensive experiments on two benchmark datasets: arXiv and PubMed (Cohan et al., 2018). The arXiv dataset contains papers in scientific domains, while the PubMed dataset contains scientific papers from the biomedical domain. These two benchmark datasets are widely adopted in long document summarization research and we use the original train, validation, and testing splits as in (Cohan et al., 2018). The detailed statistics of datasets are shown in Table 1.

Evaluation Following the common setting, we use ROUGE F-scores (Lin, 2004) as the automatic evaluation metrics. Specifically, we report the ROUGE-1/2 scores to measure summary informativeness and ROUGE-L scores to measure summary fluency. Following prior work (Liu and Lapata, 2019; Nallapati et al., 2016b), we also construct extractive ground truth labels (ORACLE) for training by greedily optimizing the ROUGE score on gold-reference abstracts.

4.2 Implementation Details

We use the publicly released BERT-base¹ (Devlin et al., 2018) as the sentence encoder. The BERT encoder is only used to generate initial sentence embeddings, but is not updated during training to improve model efficiency. We adopt the Graph Attention Network² (Veličković et al., 2017) implementation with 8 attention heads and 2 stack layers for graph message passing. The hidden size of our model is set to 2048.

Our model is trained with the Adam optimizer (Loshchilov and Hutter, 2017) with a learning rate of 0.0001 and a dropout rate of 0.1. We train our model on a single RTX A6000 GPU for 10 epochs and validate after each epoch using ROUGE-1 F-score. We employ early stopping to select the best model for a patient duration of 3 epochs. We searched the training loss re-scale factor λ in the range of 0 to 1 with 0.1 step size and got the best value of 0.5.

4.3 Baseline Methods

We perform a systematic comparison with recent approaches in both extractive and abstractive summarization for completeness. We keep the same train/validation/test splitting in all the experiments and report ROUGE scores from the original papers if available, or scores from (Xiao and Carenini, 2019) otherwise. Specifically, we compare with the following strong baseline approaches:

Unsupervised methods: LEAD method that selects the first few sentences as a summary, SumBasic (Vanderwende et al., 2007), graph-based unsupervised models LexRank (Erkan and Radev, 2004), PACSUM (Zheng and Lapata, 2019) and HIPORANK (Dong et al., 2020).

Neural extractive models: encoder-decoder based model Cheng&Lapata (Cheng and Lapata, 2016) and SummaRuNNer (Nallapati et al., 2016a); local and global context model ExtSum-LG (Xiao and Carenini, 2019) and its variants ExtSum-LG+RdLoss/MMR (Xiao and Carenini, 2020); language model-based methods SentCLF and SentPTR (Subramanian et al., 2019).

Neural abstractive models: pointer network generation model PGN (See et al., 2017), hierarchical attention generation model DiscourseAware (Cohan et al., 2018), and transformer-based generation model TLM-I+E (Subramanian et al., 2019).

¹<https://github.com/google-research/bert>

²<https://github.com/PetarV-/GAT>

	PubMed			ArXiv		
Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Oracle(15k tok.)	53.04	29.08	48.31	53.58	26.19	47.76
Lead-10	38.59	13.05	34.81	37.37	10.85	33.17
LexRank (2004)	39.19	13.89	34.59	33.85	10.73	28.99
SumBasic (2007)	37.15	11.36	33.43	29.47	6.95	26.30
PACSUM (2019)	39.79	14.00	36.09	38.57	10.93	34.33
HIPORANK (2021)	43.58	17.00	39.31	39.34	12.56	34.89
Cheng&Lapata (2016)	43.89	18.53	30.17	42.24	15.97	27.88
SummaRuNNer (2016)	43.89	18.78	30.36	42.81	16.52	28.23
ExtSum-LG (2019)	44.85	19.70	31.43	43.62	17.36	29.14
SentCLF (2020)	45.01	19.91	41.16	34.01	8.71	30.41
SentPTR (2020)	43.30	17.92	39.47	42.32	15.63	38.06
ExtSum-LG + RdLoss (2021)	45.30	20.42	40.95	44.01	17.79	39.09
ExtSum-LG + MMR (2021)	45.39	20.37	40.99	43.87	17.50	38.97
PGN (2017)	35.86	10.22	29.69	32.06	9.04	25.16
DiscourseAware (2018)	38.93	15.37	35.21	35.80	11.05	31.80
TLM-I+E (2020)	42.13	16.27	39.21	41.62	14.69	38.03
CHANGES (ours)	46.43	21.17	41.58	45.61	18.02	40.06

Table 2: ROUGE F1 results on PubMed and Arxiv datasets. We keep the same train/validation/test splitting in all the experiments and report ROUGE scores from the original papers if available, or scores from (Xiao and Carenini, 2019) otherwise.

4.4 Experiment Results

Table 2 shows the performance comparison of CHANGES and all baseline methods on both PubMed and arXiv datasets. The first blocks include the extractive ground truth ORACLE, position-based sentence selection method LEAD, and other unsupervised baseline approaches. The second block covers state-of-the-art supervised extractive neural baselines, and the third block covers the supervised abstractive baselines.

According to the results, HIPORANK (Dong et al., 2020) achieves state-of-the-art performance for graph-based unsupervised methods. Compared to PACSUM (Zheng and Lapata, 2019), the only difference is that HIPORANK incorporates section structural information for degree centrality calculation. The performance gain demonstrates the significance of capturing the hierarchical structure of academic papers when modeling cross-sentence relations.

Interestingly, the LEAD approach performs far better when summarizing short news like CNN/DailyMail (Hermann et al., 2015) and New York Times (Sandhaus, 2008) than summarizing academic papers, as shown in Table 2. The results show that the distribution of ground truth sentences in academic papers is more even. In other words, academic papers have less positional bias than news articles.

We also notice that the neural extractive models

tend to outperform the neural abstractive methods in general, possibly because the extended context is more challenging for generative models during decoding. ExtSum-LG (Xiao and Carenini, 2019) is a benchmarked extractive method with section information by incorporating both the global context of the whole document and the local context within the current topic. We argue that CHANGES could better model the complex sentence structural information with the hierarchical graph than the LSTM-minus in ExtSum-LG.

According to the experimental results, our model CHANGES outperforms all baseline approaches significantly in terms of ROUGE F1 scores on both PubMed and arXiv datasets. The performance improvements demonstrate the usefulness of the global theme-aware representations from the graph contrastive learning module and the hierarchical graph structure for identifying the salient sentences.

5 Analysis

5.1 Ablation Study

We first analyze the influence of different components of CHANGES in Table 3. Here the second row ‘w/o Contra’ means we remove the GCL module and do not update the theme-aware sentence embeddings. The third row ‘w/o Hierarchical’ denotes that we only use the theme-aware sentence embedding for prediction without hierarchical graph layers. As shown in the table, removing either

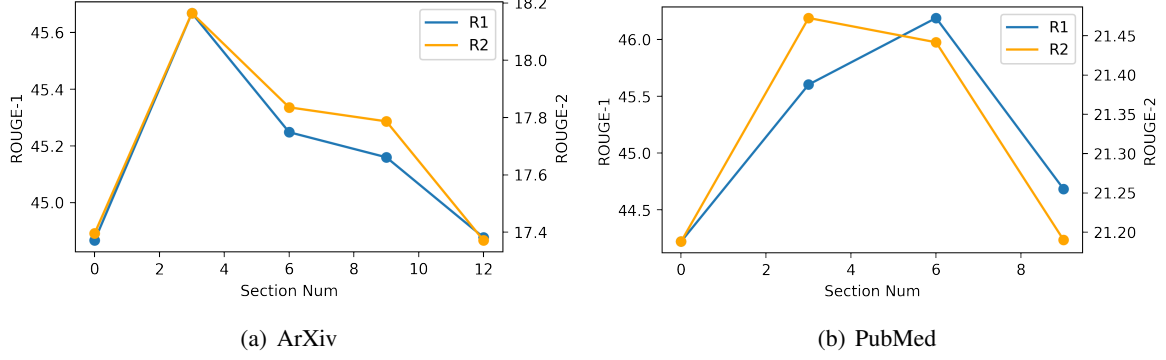


Figure 3: ROUGE-1,2 performance of CHANGES for test papers with different section numbers.

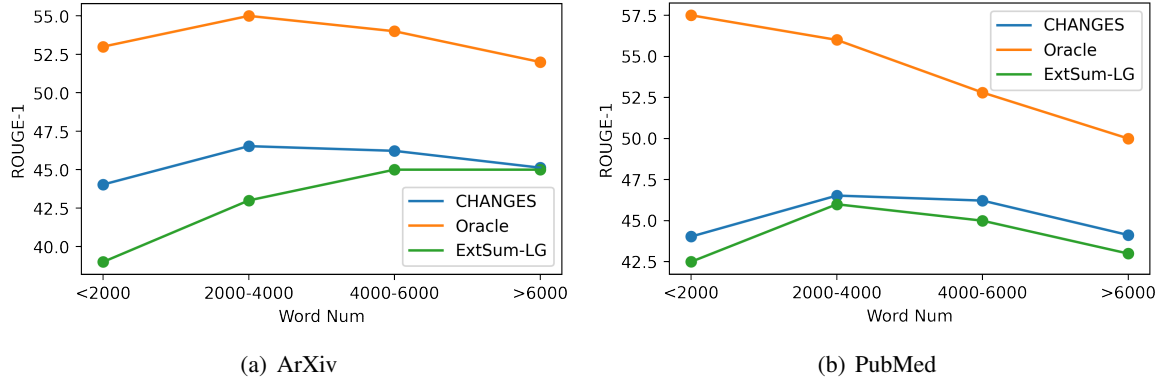


Figure 4: ROUGE-1 performance of ExtSum-LG, CHANGES, ORACLE for test papers with different lengths.

Model	ROUGE-1	ROUGE-2	ROUGE-L
PubMed			
CHANGES	46.43	21.17	41.58
w/o GCL	43.91	18.57	40.01
w/o Hierarchical	43.76	18.30	39.88
arXiv			
CHANGES	45.61	18.02	40.06
w/o GCL	44.47	16.58	38.87
w/o Hierarchical	44.72	16.79	39.10

Table 3: Ablation study results of removing components of CHANGES on PubMed and arXiv datasets.

component causes a significant model performance drop, which indicates that modeling sequential order information, semantic information, and hierarchical structural information are all necessary for academic paper summarization.

Interestingly, the theme-aware sentence embeddings and the hierarchy structure-aware sentence embeddings are almost equally critical to sentence salience modeling. The finding indicates the importance of modeling cross-sentence relations from both semantic and discourse structural perspectives.

5.2 Performance Analysis

We also analyze the sensitivity of CHANGES to section structure and length of academic papers. As shown in Figure 3, we see a performance drop trending when the number of sections increases. This is likely because the complex section structure hinders the inter-section sentence interactions. The model performance on the arXiv dataset is more stable compared to the PubMed dataset although documents in the arXiv dataset are relatively longer. We notice the same trend in Figure 4, model performance is also more stable on arXiv datasets across different document lengths. We argue this may imply that our model is more fit for longer documents that have richer discourse structures.

Regarding the document length, we see a steady performance gain when comparing to benchmark baseline methods ExtSum-LG on both datasets as shown in Figure 4. We also see as the document length increases, the performance gap between CHANGES and extractive summary performance ceiling ORACLE becomes smaller. The finding also verifies that CHANGES is especially effective and fit for long academic papers modeling.

6 Conclusion

In this paper, we propose CHANGES, a contrastive hierarchical graph-based model for scientific paper extractive summarization. CHANGES first learns global theme-aware sentence representations by graph contrastive learning module. Moreover, CHANGES incorporates the sentence-section hierarchical structure by separating intra-section and inter-section message passing and aggregating both global and local information for effective sentence embedding. Automatic evaluation on the PubMed and arXiv benchmark datasets proves the effectiveness of CHANGES and the importance of capturing both semantic and discourse structure information in modeling scientific papers.

In spite of the strong zero-shot performance of large language models like ChatGPT on various downstream tasks, long document modeling is still a challenging problem in the LLM era. Transformer-based GPT-like systems still suffer from the attention computing complexity problem and will benefit from effective and efficient modeling of long documents.

Limitations

In spite of the strong performance of CHANGES, its design still has the following limitations. First, CHANGES only extracts the sentence-section-document hierarchical structure of academic papers. We believe the model performance could be further improved by incorporating document hierarchy of different granularity like dependency parsing trees and Rhetorical structure theory trees. We leave this for future work. In addition, we only focus on single academic paper summarization in this work. Academic papers generally contain a large amount of domain knowledge, thus introducing domain knowledge from peer papers or citation networks should further boost model performance.

Acknowledgment

This work is supported by NSF through grants IIS-1763365 and IIS-2106972. We also thank the anonymous reviewers for their helpful feedback.

References

- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.
- Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.
- Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. 2022. Toward unifying text segmentation and long document summarization. *arXiv preprint arXiv:2210.16422*.
- 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*.
- Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. A supervised approach to extractive summarisation of scientific papers. *arXiv preprint arXiv:1706.03946*.
- Peng Cui and Le Hu. 2021. Topic-guided abstractive multi-document summarization. *arXiv preprint arXiv:2110.11207*.
- 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*.
- Yue Dong, Andrei Mircea, and Jackie CK Cheung. 2020. Discourse-aware unsupervised summarization of long scientific documents. *arXiv preprint arXiv:2005.00513*.
- 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.
- Marcio Fonseca, Yftah Ziser, and Shay B Cohen. 2022. Factorizing content and budget decisions in abstractive summarization of long documents by sampling summary views. *arXiv preprint arXiv:2205.12486*.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

- 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.
- Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631.
- Jiixin Ju, Ming Liu, Huan Yee Koh, Yuan Jin, Lan Du, and Shirui Pan. 2021. Leveraging information bottleneck for scientific document summarization. *arXiv preprint arXiv:2110.01280*.
- Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. [Evaluating the factual consistency of abstractive text summarization](#).
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.
- Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization. *arXiv preprint arXiv:2106.01890*.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 216–223.
- Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016a. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *arXiv preprint arXiv:1611.04230*.
- Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016b. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.
- Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636*.
- 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.
- Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.
- Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. Histruct+: Improving extractive text summarization with hierarchical structure information. *arXiv preprint arXiv:2203.09629*.
- Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.
- Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher Pal. 2019. On extractive and abstractive neural document summarization with transformer language models. *arXiv preprint arXiv:1909.03186*.
- Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- 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 Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. *arXiv preprint arXiv:2004.12393*.
- Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. *arXiv preprint arXiv:1909.08089*.
- Wen Xiao and Giuseppe Carenini. 2020. Systematically exploring redundancy reduction in summarizing long documents. *arXiv preprint arXiv:2012.00052*.

- Qianqian Xie, Jimin Huang, Tulika Saha, and Sophia Ananiadou. 2022. Gretel: Graph contrastive topic enhanced language model for long document extractive summarization. *arXiv preprint arXiv:2208.09982*.
- Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Discourse-aware neural extractive model for text summarization. *arXiv preprint arXiv:1910.14142*.
- Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.
- Michihiro Yasunaga, Rui Zhang, Kshitij Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. *arXiv preprint arXiv:1706.06681*.
- 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.
- Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2022a. Hegel: Hypergraph transformer for long document summarization. *arXiv preprint arXiv:2210.04126*.
- Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Diffusum: Generation enhanced extractive summarization with diffusion. *arXiv preprint arXiv:2305.01735*.
- Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023b. Extractive summarization via chatgpt for faithful summary generation. *arXiv preprint arXiv:2304.04193*.
- Haopeng Zhang, Semih Yavuz, Wojciech Kryscinski, Kazuma Hashimoto, and Yingbo Zhou. 2022b. Improving the faithfulness of abstractive summarization via entity coverage control. *arXiv preprint arXiv:2207.02263*.
- Haopeng Zhang and Jiawei Zhang. 2020. Text graph transformer for document classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- 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*.
- 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. *arXiv preprint arXiv:1807.02305*.