HeterGraphLongSum: Heterogeneous Graph Neural Network with Passage Aggregation for Extractive Long Document Summarization

Tuan-Anh Phan and **Ngoc-Dung Nguyen** and **Khac-Hoai Nam Bui**[*]
Viettel Cyperspace Center, Viettel Group, Vietnam
{anhpt161,dungnn7,nambkh}@viettel.com.vn

Abstract

Graph Neural Network (GNN)-based models have proven effective in various Natural Language Processing (NLP) tasks in recent years. Specifically, in the case of the Extractive Document Summarization (EDS) task, modeling documents under graph structure is able to analyze the complex relations between semantic units (e.g., word-to-word, word-to-sentence, sentence-to-sentence) and enrich valuable information for the sentence representation. However, long-form document summarization using graph-based approaches is still an open research issue. The main challenge is to represent long documents in a graph structure in an effective way. In this regard, this paper proposes a new heterogeneous graph neural network (HeterGNN) model to improve the performance of long document summarization (HeterGraphLongSum). Specifically, the main idea is to add the passage nodes into the heterogeneous graph structure of word and sentence nodes for enriching the final representation of sentences. In this regard, HeterGraphLongSum includes three types of semantic units such as word, sentence, and passage. Experiments on two benchmark datasets for long documents such as Pubmed and Arxiv indicate promising results of the proposed model for the extractive long document summarization problem. Especially, HeterGraphLongSum is able to achieve state-of-the-art performance without relying on any pre-trained language models (e.g., BERT). The source code is available for further exploitation on the Github[1].

1 Introduction

Document summarization is one of the central problems in NLP, which aims to rewrite a single document or multi documents under a shorter version with preserving the main information. There are two major approaches such as *extractive* and *abstractive* summarization. *Abstractive models* are more sophisticated abilities that require well-comprehensive reading text and generating high-quality text. Specifically, most of the existing architectures have been built based on sequence-to-sequence (Seq2Seq) techniques in different ways such as Recurrent Neural Network (RNN) (Nallapati et al., 2017), Pointer-Generator-Network(See et al., 2017), or Transformer-based models(Zhang et al., 2020; Xiao and Carenini, 2020). Furthermore, the external information, for instance, pretrained model BERTSum(Liu and Lapata, 2019) and topic modeling (Wang et al., 2020b; Nguyen et al., 2021)) are incorporated to improve performances. Nevertheless, this approach requires a complicated neural network that consists of millions of learnable parameters, which is the cause of raising significant costs in both terms of computation time perplexity and resources. Therefore, *extractive models* still gain much attention. Particularly, extractive document summarization (EDS) takes a document in the form list of sentences and chooses several best candidates from the original document, then combine them to create the summarization. Recent models trend to turn EDS into the sequential binary-labeling task (Nallapati et al., 2017; Cheng and Lapata, 2016; Zhou et al., 2018).

Graph neural network (GNN) has recently been exploited as an emerging line of deep learning architectures, which has powered various domains, including NLP tasks (Vashishth et al., 2020). Specifically, GNN models are able to model complex structural data containing semantic units (node) with relationships (edge) between them (Xu et al., 2019). For the EDS task, each document is represented as a graph structure in which the nodes are the semantic units of the document such as words and sentences. Sequentially, developing edges among sentence nodes are capable to model the cross-sentence relations, which is able to handle the limitation of traditional Seq2Seq-based meth-

[*] corresponding author

[1] https://github.com/tuananhphan97vn/HeterGraphLongSum

Proceedings of the 29th International Conference on Computational Linguistics, pages 6248–6258
October 12–17, 2022.

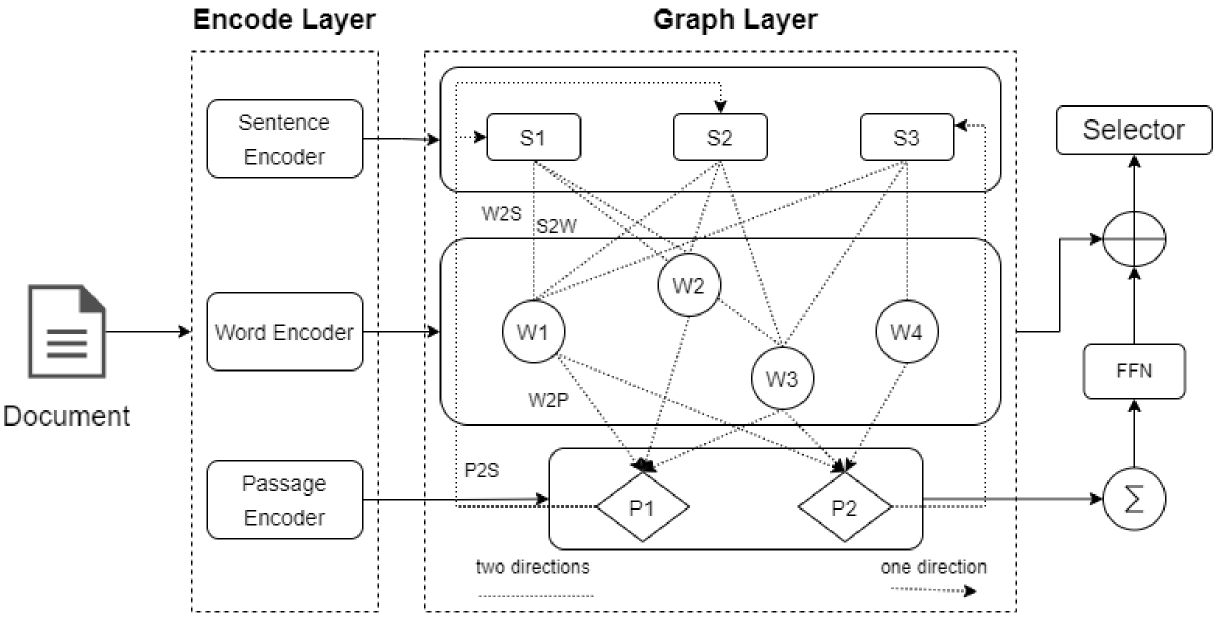


Figure 1: Overview of HeterGraphLongSum model. Passages of each document are defined as a set of sentences in sequence with a fixed number of sentences. In this architecture, the edges from *passage to word* and *sentence to passage* are not taken into account because of the redundancy.

ods in terms of long-distance dependency among sentences (Cui et al., 2020). In particular, cross-sentence edges can be constructed explicitly between sentences (sentence-to-sentence) (Xiao and Carenini, 2019; Jing et al., 2021; Yasunaga et al., 2017) or through intermediate bridge via common words (sentence-word-sentence)(Wang et al., 2020a) or latent topics (sentence-topic-sentence) (Cui et al., 2020).

Although the aforementioned approaches have achieved remarkable results in the EDS problem, most of the architectures are proposed for short documents (i.e., new articles). Long-form document is still a remaining challenge in this research field due to two main reasons: i) most traditional Seq2Seq methods truncate longer documents into small fixed-length sequences (i.e., passages) (Zaheer et al., 2020; Zhang et al., 2021), which leads to information loss problem, especially for the extractive summarization; ii) using GNN-based methods is able to mitigate the information loss by enabling cross-sentence relations, however, representing an effective way for long-text documents into graph structure is still an open research issue. Specifically, since the vocabulary size is limited, when the length of the document is increased, more sentences become neighbors with each other (via common words) which is the cause of the similar embedding between sentences. Therefore, a graph structure, which includes only word nodes and sentence nodes, might not be an effective way to represent the long documents for the EDS problem.

In order to alleviate the aforementioned challenges, this paper presents a new graph-based architecture, which contains three semantic units such as word, sentence, and passage. In particular, the passage nodes are adopted for learning the cross-relations between sentences in different passages. Furthermore, the passage node can be regarded as the local structure of a group of sentence nodes in which the edges between passages and sentences have the possibility to reduce the harm of similar representations of sentences when expanding graph structure with long documents. Figure 1 illustrates the model architecture of HeterGraphLongSum. Specifically, the main contributions of this paper are threefold as follows:

- We present a novel GNN-based method for modeling long-form documents. Specifically, instead of using common methods for learning long documents with the hierarchical perspective (e.g., word-to-sentence-to-passage), we consider passage as one of the node types, which is updated simultaneously with other nodes in the graph. In this regard, more semantic units (additional nodes) in the graph enable the capability to enrich the cross-relations between elements (e.g., sentence representation).

- We propose a new Heterogeneous GNN (HeterGNN) model for the EDS task, focusing on long documents (e.g., scientific papers). Especially, we consider this issue without employing pre-trained encoders (e.g., BERT). In this regard, our method is able to extend to other low-resource languages without any obstacles.

- We evaluate the proposed model with two benchmark long document datasets such as PubMed and ArXiv. The experiential results indicate that our method is able to achieve the state-of-the-art level in this research field.

2 Related Work

2.1 Neural Extractive Summarization

Neural networks have achieved great success in extractive summarization, which explores different neural components to develop an end-to-end learning model (Zhong et al., 2019). The encoder-decoder frameworks are mainly developed by using RNN (Cheng and Lapata, 2016; Nallapati et al., 2017; Zhou et al., 2018) and Transformer (Zhang et al., 2020; Xiao and Carenini, 2020) with auto-regressive (Jadhav and Rajan, 2018; Liu and Lapata, 2019) or non auto-regressive (Narayan et al., 2018; Arumae and Liu, 2018) decoder. Sequentially, recent remarkable results are mainly developed by using pre-trained language models (e.g., BERT (Devlin et al., 2019)) such as BERTSUM (Liu and Lapata, 2019) and MATCHSUM (Zhong et al., 2020). Most of the aforementioned studies formulate the EDS task as sentence labeling or sentence ranking problems. In this paper, we formulated this task as the binary-labeling problem (Nallapati et al., 2017) and exploited our model with a non-pre-trained CNN/BiLSTM encoder in which we believe that this method is able to easily extend to other low resource languages.

2.2 Graph-based Summarization

Early works on graph-based methods for EDS tasks rely on the similarity scores between sentences in unsupervised manners such as TextRank (Mihalcea and Tarau, 2004) and LexRank (Erkan and Radev, 2004). The core idea of using graph representation is to utilize the linguistic information of sentences. Consequentially, GNNs have been adopted for learning cross-sentence relations with remarkable performances, using the concept of discourse graph (Yasunaga et al., 2017; Xu et al., 2020). Recently, the trend research focuses on representing documents with different types of nodes (heterogeneous graphs) to utilize the effects of additional semantic units such as words, sentences (Wang et al., 2020a; Jin et al., 2020) and latent topics (Cui et al., 2020). In this study, the proposed model exploits the heterogeneous graph structure with more complex units by adding semantic passage nodes to leverage the problems of adopting graph-based models in long document summarization.

2.3 Long Document Summarization

Long document summarization has recently received increased attention since the remained challenge of modeling long texts (Fremann and Klementiev, 2019). Specifically, the current potential solution for this issue is to truncate documents into small fixed-length sequences and use sliding window methods to process the document separately (Beltagy et al., 2020; Zaheer et al., 2020). However, this paradigm leads the serious information loss, which is not suitable for the EDS task, due to this task requiring the information relations of extracted sentences (Li et al., 2020). In this regard, several promising approaches have been introduced for long document summarization. Cohan et al. (2018) presents a hierarchical encoder to capture the discourse structure of the input document with a discourse-aware decoder for abstractive summarization. Xiao and Carenini (2019) leverages the long text summarization task by incorporating a distributed representation of both the global (whole document) and local (section/topic) contexts. Cui and Hu (2021) employs a dynamic memory network with sliding multiple windows to mitigate the information loss between segments of sentences. Regarding the graph-based methods, Cui et al. (2020) adopts a modified graph attention network (GAT) for capturing inter-sentence relationship. Furthermore, latent topics are added as an additional type of node, which incorporates sentence nodes to improve the performance of long document summarization in terms of capturing the relational information of long-distance sentences. In this study, we take the graph-based structure for EDS of long text into account with a different perspective by considering word nodes and sentence nodes for capturing both inter and intrasentence relations. Moreover, passage nodes are jointly trained to improve the performance of long texts by learning the cross-relations of sentences with long-distance and mitigating the similar representation problem in the large-scale graph structure.

3 HeterGraphLongSum model

HeterGraphLongSum aims to learn a heterogeneous graph structure for long text summarization. Specifically, we model an input document with three types of nodes such as *word*, *sentence*, and *passage* nodes, as a heterogeneous graph, and using graph attention network (GAT) (Velickovic et al., 2017) for capturing information relations among nodes.

3.1 Graph Construction

Let $G = \{V, E\}$ represent an arbitrary graph, where V and E denote the node and edge sets, respectively. Specifically, as shown in the Figure 1, our directed graph can be defined as $V = \{V_w \cup V_s \cup V_p\}$ and $E = \{E_{w2s} \cup E_{s2w} \cup E_{w2p} \cup E_{p2s}\}$, where V_w, V_s, and V_p stand for three semantic units of a document (i.e., word, sentence, and passage), and E_{w2s}, E_{s2w}, E_{w2p}, and E_{p2s} stand for four types of edges such as *word-to-sentence*, *sentence-to-word*, *word-to-passage*, and *passage-to-sentence*, respectively. Accordingly, the proposed heterogeneous graph structure is designed based on two assumptions as follows:

- The passage units are not available on most publicity datasets in this research field. Therefore, following the previous works on long-form document representations(Zaheer et al., 2020; Zhang et al., 2021), we format the passages in form of a sequence of sentences and created them by concatenating a fixed size with n sentences. In this regard, the number of sentences for each passage is a hyperparameter, which is tuned during the validation process.

- Regarding the certain edge types, instead of adopting the full connection between semantic units, only four types of edges are taken into account such as *word-to-sentence* (w2s), *sentence-to-word* (s2w), *word-to-passage* (w2p), and *passage-to-sentence* (p2s). Accordingly, the edge from *passage-to-word* (p2w) and *sentence-to-passage* (s2p) are not considered because of the redundancy. Specifically, $p2w$ is not considered since many words receive the same information (i.e., from the passage), which might harm the overall performance. Furthermore, there are two types of edge to update the passage information such as $w2p$ and $s2p$. In this regard, we design $w2p$ in our graph structure to enable the cross-passage relations via path $passage \rightarrow sentence \rightarrow word \rightarrow passage$ and remove the $s2p$ edge type. We prove this assumption via the ablation study in the experiment section.

Intuitively, by adding two types of edges from passage nodes, cross-sentences relations can be simultaneously processed in two ways: i) *local information* with path $sentence \rightarrow word \rightarrow passage \rightarrow sentence$; ii) *global information* with path $sentence \rightarrow word \rightarrow sentence$. The additional *local information* enables the model to mitigate the problem of similarity representation between sentences when the graph structure is expanded by adding passage information. Specifically, this issue is specific to sentences located in different positions in the document, which is especially suitable for learning long documents.

3.2 Graph Encoder Embedding

Supporting the matrix features of word node, sentence node and passage node are sequentially denoted as $X_w \in R^{|V_w| \times d_w}, X_s \in R^{|V_s| \times d_s}$, and $X_p \in R^{|V_p| \times d_p}$, respectively. The initialized embedding representation of the word node is encoded by using Glove (Pennington et al., 2014). In the case of sentences, instead of using pre-trained models, we combine Convolutional Neural Network (CNN) and bidirectional Long Short-Term Memory (BiLSTM) for the encoder, which can be formulated as follows:

$$(X_s)_j = CNN(x_{1:m}) \oplus BiLSTM(x_{1:m}) \quad (1)$$

where m denotes the number of words in the sentence s_j. In this regard, the Passage feature is encoded by using Bi-directional LSTM based on the hidden state of sentences, which is extracted from the last layer network as follows:

$$(X_p)_i = BiLSTM\left((X_s)_{(j)}\right) \quad (2)$$

where $(X_p)_i$ denotes embedding of the *i-th* passage node, j and k are the *j-th* sentence and number of sentence per passage ($k * i \leq j \leq k * (i + 1)$), respectively.

3.3 Graph Learning Layer

The vectors of nodes are initialized with embedding features, where $H_s^0 = X_s$ $H_w^0 = X_w$, and $H_p^0 = X_p$, respectively. Sequentially, the node representations are updated with the graph attention network.

Graph Attention Network: Given the heterogeneous graph structure and initialized features of each node, GAT is adopted to calculate the hidden states of nodes. Specifically, supporting $\vec{h_i} \in R^{d_{h_i}}$ and N_i denote the input hidden representation and the neighbors of node *i-th*, respectively, the graph attention layer can be calculated as follows:

$$\begin{aligned} z_{ij} &= LeakyRelu(\vec{a}^T(W_q\vec{h_i}||W_k\vec{h_j})) \\ \alpha_{ij} &= \frac{e^{z_{ij}}}{\sum_{k \in N_i} e^{z_{ik}}} \\ \vec{h_i'} &= \sigma\left(\sum_{j \in N_i} \alpha_{ij} W_v \vec{h_j}\right) \end{aligned} \tag{3}$$

where W_q, W_k, W_v, and $\vec{\alpha}$ are learnable parameters and optimized during the training process. The symbol $||$ indicates the concatenation operator. σ denotes the non-linear transform function and $\vec{h_i'}$ denotes the hidden state which presents information gained from the neighboring nodes. Alternatively, multi-head attention can be used for improving the performance, which is calculated as follows:

$$\vec{h_i'} = ||_{k=1}^K \, \sigma\left(\sum_{j \in N_i} \alpha_{ij} W_v^k \vec{h_j}\right) \tag{4}$$

Furthermore, in order to mitigate the gradient vanishing problem, the residual connection is added to the original representation to provide the final hidden state:

$$\vec{h_i''} = \vec{h_i} + \vec{h_i'} \tag{5}$$

Graph Propagation: After initialization, the sentence nodes are updated with their neighbor word nodes and passage nodes by using GAT and FFN layer:

$$\begin{aligned} U_{w2s}^1 &= GAT(H_s^0, H_w^0, H_w^0) \\ U_{p2s}^1 &= GAT(H_s^0, H_p^0, H_p^0) \\ U_s^1 &= \sigma(U_{w2s}^1 + U_{p2s}^1) \\ H_s^1 &= FFN(U_s^1 + H_s^0) \end{aligned} \tag{6}$$

Sequentially, word nodes are updated with the new representation of sentences. Similarly, the passage nodes are updated by the updated word embedding. The updated process at an iteration of GAT can be formulated as follows:

$$\begin{aligned} U_{w2s}^t &= GAT(H_s^{t-1}, H_w^{t-1}, H_w^{t-1}) \\ U_{p2s}^t &= GAT(H_s^{t-1}, H_p^{t-1}, H_p^{t-1}) \\ U_s^t &= \sigma(U_{w2s}^t + U_{p2s}^t) \\ H_s^t &= FFN(U_s^t + H_s^{t-1}) \\ \\ U_w^t &= GAT(H_w^{t-1}, H_s^t, H_s^t) \\ H_w^t &= FFN(U_w^t + H_w^{t-1}) \\ \\ U_p^t &= GAT(H_p^{t-1}, H_w^t, H_w^t) \\ H_p^t &= FFN(U_p^t + H_p^{t-1}) \end{aligned} \tag{7}$$

Note that, H_w^1 and H_p^1 are set to the same values with H_w^0 and H_p^0, respectively.

3.4 Sentence Extraction

For the sentence selector layer, we first extract document representation from the hidden state of passages via the attention layer, then combine document representation and each sentence by using the concatenate operator, which is sequentially formulated as follows:

$$\begin{aligned} z_i &= ReLu(\vec{a_p^T}(\vec{h_p})_i) \\ \alpha_i &= \frac{e^{z_i}}{\sum_j e^{z_j}} \\ \vec{h_d} &= \sum_m \alpha_m * \left(\vec{h_p}\right)_m \end{aligned} \tag{8}$$

$$\vec{h_{d,s_k}} = FFN\left(\vec{h_d} \parallel \vec{h_{s_k}}\right) \tag{9}$$

where $\vec{a_p}$ is learnable parameter. i and k represent indexes of passage *i-th* and sentence *k-th*, respectively. α_i indicates the amount of contribution of passage *i-th* to document representation $\vec{h_d}$. Consequently, if α_i gets a high attention score, this passage tends to be more significant than other passages. Finally, the output sentence-document representation $\vec{h_{d,s_i}}$ is used for sentences classification by using binary cross-entropy loss as the training objective:

$$L = \frac{1}{N} \sum_{i=1}^{i=N} y_i log\left(\hat{y_i}\right) + (1 - y_i) \, log\left(1 - \hat{y_i}\right) \tag{10}$$

Model	arXiv			PubMed		
	R-1	R-2	R-L	R-1	R-2	R-L
SumBasic*	29.47	6.95	26.30	37.15	11.36	33.43
LexRank*	33.85	17.36	28.99	39.19	13.89	34.59
Oracle$^+$	53.88	23.05	34.90	55.05	27.48	38.66
Cheng & Lapata (2016)$^+$	42.24	15.97	27.88	43.89	18.53	30.17
SummaRuNNer$^+$	42.91	16.65	28.53	43.89	18.78	30.36
Xiao & Carenini (2019)(Xiao and Carenini, 2019)	43.62	17.36	29.14	44.85	19.7	31.43
Match-Sum	40.59	12.98	32.64	41.21	14.91	36.75
Topic-GraphSum(Cui et al., 2020)	44.03	18.52	32.41	45.95	20.81	33.97
SSN-DM(Cui and Hu, 2021)	45.03	19.03	32.58	46.73	21.00	34.10
HeterGraphLongSum (iter=1)	46.62	18.69	40.77	48.75	22.45	43.97
HeterGraphLongSum (iter=2)	**47.36**	**19.11**	**41.47**	**48.86**	**22.63**	**44.19**

Table 1: Results on the test set. Report results with * are from Cohan et al. (2018), and results with + are from Xiao and Carenini (2019). Other results are obtained from respective papers. Our results are calculated by averaging values of 3 runs.

4 Experiment

4.1 Experimental setup

Dataset: two benchmark datasets of long documents are considered for the experiments such as arXiv and PubMed datasets, which are scientific papers. Accordingly, those datasets are processed following the work in Cohan et al. (2018) and get Oracle results, a gold standard extractive label, by the work in Xiao and Carenini (2019). The statistics of evaluated datasets are illustrated in Table 2.

Datasets	Documents			Avg. Tokens	
	Train	Val	Test	Doc.	Sum.
arXiv	203,037	6,436	6,440	4,938	220
PubMed	119,924	6,633	6,658	3,016	203

Table 2: Statistics of experiential datasets.

Models for comparision: we evaluate the proposed models with recent benchmark models in this research field which are mainly divided into four approaches: traditional EDS models such as SumBasic (Vanderwende et al., 2007) and LexRank (Erkan and Radev, 2004); Seq2Seq-based models such as Cheng & Lapata (Cheng and Lapata, 2016), SummaRuNNer (Nallapati et al., 2017), and Xiao & Carenini (Xiao and Carenini, 2019); pretrained-based models such as Match-Sum (Zhong et al., 2020); graph-based models such as Topic-GraphSum(Cui et al., 2020) and SSN-DM (Cui and Hu, 2021).

Hyperparameter setting: Regarding the word node generation, following previous work (Xiao and Carenini, 2019), the vocabulary is limited to 50,000. The word embedding initializes with 100 dimensions using Glove pre-trained model (Pennington et al., 2014). The dimension of the sentence and passage are both set to 64. The dimension of the final output representation of all models is set to 64. The multi-head of the GAT layer for s2w is set to 4 and others (i.e., w2s, w2p, and p2s) are set to 1. The passage length is a hyperparameter, which is a constant number. Specifically, we vary the value of passage length from 10 to 30 in order to determine the best results for two evaluated datasets. More details of the impact of passage length are presented in the ablation section. We select top-6 of PubMed and top-5 of arXiv datasets for the decoding process, according to the best performance of the validation set. All models are trained for 20 epochs with a single NVIDIA V100 card (batch size = 32) and use early stopping on the validation set according to entropy loss in order to select the best model.

4.2 Main Results

Table 1 reports the evaluation results on two benchmark datasets. Accordingly, ROUGE is used as the evaluation metric, which includes unigram (R-1), bigram (R-2) overlap, and longest common subsequence (R-L) for measuring informativeness and assessing fluency, respectively. The results are presented in different sections corresponding to different approaches. The first section includes traditional approaches and the Oracle. The second section obtains the results of Seq2Seq-based mod-

Our Model	arXiv			PubMed		
	R-1	R-2	R-L	R-1	R-2	R-L
w/o Passage Node	46.43	18.62	40.56	47.81	21.88	43.01
w/o Doc. Rep.	46.61	18.8	40.76	48.52	22.34	43.77
Proposed Model	47.05	19.01	41.2	48.86	22.63	44.19

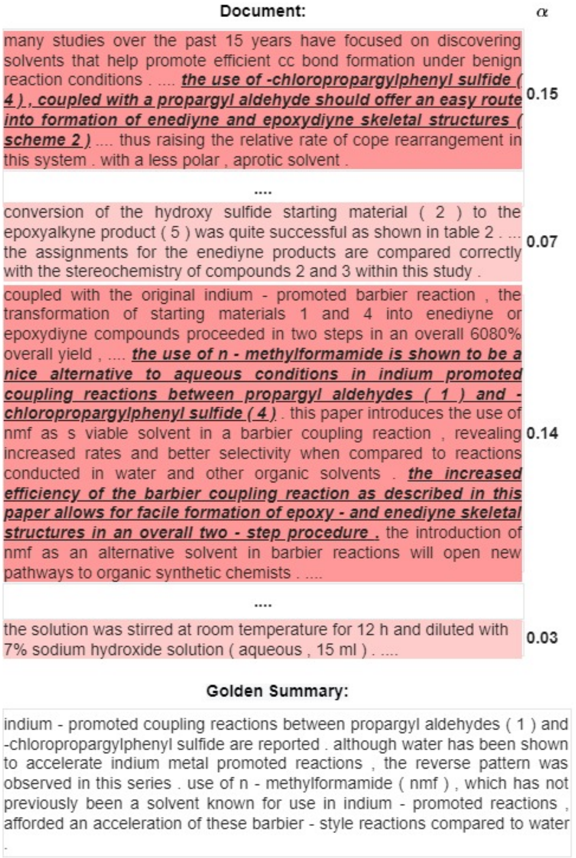
Table 3: Reported results of our proposed model and two ablated variants on two benchmark datasets.

els. The third section is Match-Sum, a state-of-the-art BERT-based summarizing model. The next section reports recent graph-based models for the EDS problem. The last section is our model, which includes two versions with different iterations of GAT layers.

Based on the evaluation results, several hypotheses for extractive long document summarization problem can be expressed as follows: i) Using pre-trained models (e.g., BERT and RoBERTa) without any modifications is not effective for long documents. The main reason is the limitation of 512 tokens of BERT-based models; ii) Exploiting the global context (the whole document) is able to improve the performance, even without the need for pre-trained models for extracting features; iii) Using graph layer with external information (e.g., latent topic) for encoder embedding is currently state-of-the-art approach in this research field; iv) our model, which incorporates global context with graph neural network, achieve state-of-the-art results on both benchmark datasets of long documents. Especially, the most advantage is that our method provides promising performances without external information and pre-trained language models. A limitation of our study is that we use the fixed length of sentences for passages, which might not suitable for all datasets with the same value (more detail in the ablation study section). An appropriate solution is to adopt semantic self-segmentation methods for determining passages (Moro and Ragazzi, 2022). We leave this issue for future work of this study.

4.3 Ablation Study

Ablated Variants: in order to analyze the impact of each module in the proposed architecture, we evaluate the proposed model with two ablated variants such as i) **w/o Passage Node** removes the passage node in the heterogeneous graph structure, build a HeterGNN with two types of nodes such as word and sentence; ii) **w/o Doc. Rep.** removes the document representation from passages for the sentence extraction process (Eq. 8). Table 3 shows the results of different variants on two evaluated datasets. As result, the proposed architecture outperforms all variants, which proves that combining both modules can achieve the best results. Especially, by adding passage nodes, the similar representation problem of sentences, when the nodes of words and sentences are increased to represent the long-form document, can be reduced. In particular, the effectiveness of proposed modules is visualized with an example in Figure 2. Accordingly,



Document:

α

many studies over the past 15 years have focused on discovering solvents that help promote efficient cc bond formation under benign reaction conditions . .... ***the use of -chloropropargylphenyl sulfide (4) , coupled with a propargyl aldehyde should offer an easy route into formation of enediyne and epoxydiyne skeletal structures (scheme 2)*** .... thus raising the relative rate of cope rearrangement in this system . with a less polar , aprotic solvent . 0.15

....

conversion of the hydroxy sulfide starting material (2) to the epoxyalkyne product (5) was quite successful as shown in table 2 . ... the assignments for the enediyne products are compared correctly with the stereochemistry of compounds 2 and 3 within this study . 0.07

coupled with the original indium - promoted barbier reaction , the transformation of starting materials 1 and 4 into enediyne or epoxydiyne compounds proceeded in two steps in an overall 6080% overall yield , ... ***the use of n - methylformamide is shown to be a nice alternative to aqueous conditions in indium promoted coupling reactions between propargyl aldehydes (1) and -chloropropargylphenyl sulfide (4)*** . this paper introduces the use of nmf as s viable solvent in a barbier coupling reaction , revealing increased rates and better selectivity when compared to reactions conducted in water and other organic solvents . ***the increased efficiency of the barbier coupling reaction as described in this paper allows for facile formation of epoxy - and enediyne skeletal structures in an overall two - step procedure .*** the introduction of nmf as an alternative solvent in barbier reactions will open new pathways to organic synthetic chemists 0.14

....

the solution was stirred at room temperature for 12 h and diluted with 7% sodium hydroxide solution (aqueous , 15 ml) 0.03

Golden Summary:

indium - promoted coupling reactions between propargyl aldehydes (1) and -chloropropargylphenyl sulfide are reported . although water has been shown to accelerate indium metal promoted reactions , the reverse pattern was observed in this series . use of n - methylformamide (nmf) , which has not previously been a solvent known for use in indium - promoted reactions , afforded an acceleration of these barbier - style reactions compared to water .

Figure 2: Visualized the efficiency of using passage nodes to enhance sentence representation. The degree of highlighting expresses the important role of the passage in the document. Underlined sentences are model-selected summaries. As result, the selected sentences belong to passages that have high scores of α (Equation 8).

Our Model	arXiv			PubMed		
	R-1	R-2	R-L	R-1	R-2	R-L
w/o w2p	46.30	18.50	40.38	48.76	22.57	44.08
w/o p2s	46.87	18.84	40.96	48.76	22.45	44.00
plus p2w	46.77	18.82	40.89	48.80	22.54	44.06
plus s2p	46.62	18.73	40.70	48.23	22.06	43.43
full edge	46.92	18.80	41.07	48.51	22.42	43.75
Proposed Model	47.05	19.01	41.20	48.86	22.63	44.19

Table 4: Evaluation on the impact of edge types.

each passage has different impacts on the document representation and can be utilized effectively to enrich the information of sentence representations. As shown in the example, the selected sentences mainly belong to the passages, which have high attention scores (Equation 8).

Impact of Edge Types: the proposed heterogeneous graph includes four types of edge such as *s2w*, *w2s*, *w2p*, and *p2s*. Accordingly, *p2w* and *s2p* are removed because of redundancy, which might influence the performance. In this section, we try to evaluate the impact of edge types on the performance by developing several variants of the proposed graph structure, which are: i) **w/o w2p** removes the link from word nodes to passage nodes; ii) **w/o p2s** removes the link from passage nodes to sentence nodes; iii) **plus p2w** adds the link from passage nodes to word nodes; iv) **plus s2p** adds the link from sentence nodes to passage nodes; and v) **full edge** builds a HeterGNN model of three types of nodes such as word, sentence and passage nodes with fully connected among nodes. Specifically, there are total six types of edges of this model such as *w2s*, *w2p*, *s2w*, *s2p*, *p2w*, and *p2s*. Table 4 shows the results of the evaluation. Accordingly, the results indicate that using four types of edges in the proposed model is able to achieve the best results for both evaluated datasets. Note that all the ablated experiments use the same value of passage length (n= 10). More details about this hyperparameter are exploited in the following section.

Length of Passage: is an important hyperparameter in this study in which different datasets might require different values of passage length. In this regard, we conduct experiments to determine the best values of passage length for two evaluated datasets. Table 5 illustrates the impact of passage length on the performance of two datasets, respectively. Specifically, the value of passage length is ranged from 10 to 30 (per 05 periods). As result, the best values of passage length for arXiv and PubMed are 30 and 10, respectively. The experimented result indicates a hypothesis that an adaptive method for automatically segmenting the passage length is able to improve performance. In particular, the passage can be segmented by unsupervised (Alemi and Ginsparg, 2015) or supervised (Koshorek et al., 2018) learning. We take this issue into account for the future work of this study.

Datasets	n	R-1	R-2	R-L
arXiv	10	47.05	19.01	41.20
	15	46.68	18.79	40.78
	20	46.14	18.47	40.30
	25	46.81	18.79	40.95
	30	**47.36**	**19.11**	**41.47**
PubMed	10	**48.86**	**22.63**	**44.19**
	15	48.53	22.26	43.75
	20	48.75	22.45	43.97
	25	48.75	22.57	44.02
	30	**48.86**	22.45	44.07

Table 5: Impact of passage length on the performances of the proposed model.

5 Conclusion

This paper presents a new GNN-based model for extractive long document summarization. Specifically, GNN has been introduced as a promising approach for exploiting the complex relation of elements (e.g., word and sentence) from an input document. However, representing long documents as graph structure is still a remaining challenge. Specifically, lacking cross-relation information between sentences (e.g., long-distance of position in the document) and the increment of nodes might influence the performance. In this regard, this paper proposes a heterogeneous graph including three types of nodes such as word, sentence, and passage, which are simultaneously learned for enabling the cross-relation between sentences. The evaluation of two standard long documents datasets such as arXiv and PubMed shows that the proposed model outperforms state-of-the-art models in this research field without relying on pre-trained language models (e.g., BERT).

References

Alexander A. Alemi and Paul Ginsparg. 2015. Text segmentation based on semantic word embeddings. *CoRR*, abs/1503.05543.

Kristjan Arumae and Fei Liu. 2018. Reinforced extractive summarization with question-focused rewards. In *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, Student Research Workshop*, pages 105–111. Association for Computational Linguistics.

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

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 615–621. Association for Computational Linguistics.

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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5881–5891. Association for Computational Linguistics.

Peng Cui, Le Hu, and Yuanchao Liu. 2020. Enhancing extractive text summarization with topic-aware graph neural networks. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5360–5371. International Committee on Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.*, 22:457–479.

Lea Frermann and Alexandre Klementiev. 2019. Inducing document structure for aspect-based summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6263–6273. Association for Computational Linguistics.

Aishwarya Jadhav and Vaibhav Rajan. 2018. Extractive summarization with SWAP-NET: sentences and words from alternating pointer networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 142–151. Association for Computational Linguistics.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6244–6254. Association for Computational Linguistics.

Baoyu Jing, Zeyu You, Tao Yang, Wei Fan, and Hanghang Tong. 2021. Multiplex graph neural network for extractive text summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 133–139. Association for Computational Linguistics.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text segmentation as a supervised learning task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 469–473. Association for Computational Linguistics.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6232–6243. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3728–3738. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 404–411. ACL.

Gianluca Moro and Luca Ragazzi. 2022. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11085–11093.

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 Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3075–3081. AAAI Press.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1747–1759. Association for Computational Linguistics.

Thong Nguyen, Anh Tuan Luu, Truc Lu, and Tho Quan. 2021. Enriching and controlling global semantics for text summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9443–9456. Association for Computational Linguistics.

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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Inf. Process. Manag.*, 43(6):1606–1618.

Shikhar Vashishth, Naganand Yadati, and Partha P. Talukdar. 2020. Graph-based deep learning in natural language processing. In *CoDS-COMAD 2020: 7th ACM IKDD CoDS and 25th COMAD, Hyderabad India, January 5-7, 2020*, pages 371–372. ACM.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *CoRR*, abs/1710.10903.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020a. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6209–6219. Association for Computational Linguistics.

Zhengjue Wang, Zhibin Duan, Hao Zhang, Chaojie Wang, Long Tian, Bo Chen, and Mingyuan Zhou. 2020b. Friendly topic assistant for transformer based abstractive summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 485–497. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3009–3019. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2020. Systematically exploring redundancy reduction in summarizing long documents. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 516–528. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5021–5031. Association for Computational Linguistics.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir R. Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 452–462. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hang Zhang, Yeyun Gong, Yelong Shen, Weisheng Li, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2021. Poolingformer: Long document modeling with pooling attention. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12437–12446. PMLR.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6197–6208. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1049–1058. Association for Computational Linguistics.

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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 654–663. Association for Computational Linguistics.