electronics

Article

A Multi-Granularity Heterogeneous Graph for Extractive Text Summarization

Henghui Zhao [1], Wensheng Zhang [1,2,*], Mengxing Huang [1,*], Siling Feng [1] and Yuanyuan Wu [1]

[1] School of Information and Communication Engineering, Hainan University, Haikou 570100, China
[2] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
* Correspondence: wensheng.zhang@ia.ac.cn (W.Z.); huangmx09@hainanu.edu.cn (M.H.)

Abstract: Extractive text summarization selects the most important sentences from a document, preserves their original meaning, and produces an objective and fact-based summary. It is faster and less computationally intensive than abstract summarization techniques. Learning cross-sentence relationships is crucial for extractive text summarization. However, most of the language models currently in use process text data sequentially, which makes it difficult to capture such inter-sentence relations, especially in long documents. This paper proposes an extractive summarization model based on the graph neural network (GNN) to address this problem. The model effectively represents cross-sentence relationships using a graph-structured document representation. In addition to sentence nodes, we introduce two nodes with different granularity in the graph structure, words and topics, which bring different levels of semantic information. The node representations are updated by the graph attention network (GAT). The final summary is obtained using the binary classification of the sentence nodes. Our text summarization method was demonstrated to be highly effective, as supported by the results of our experiments on the CNN/DM and NYT datasets. To be specific, our approach outperformed baseline models of the same type in terms of ROUGE scores on both datasets, indicating the potential of our proposed model for enhancing text summarization tasks.

Keywords: graph neural network; heterogeneous graph; attention mechanism; implicit topic

Citation: Zhao, H.; Zhang, W.; Huang, M.; Feng, S.; Wu, Y. A Multi-Granularity Heterogeneous Graph for Extractive Text Summarization. *Electronics* **2023**, *12*, 2184. https://doi.org/10.3390/electronics12102184

Academic Editor: Arkaitz Zubiaga

Received: 5 April 2023
Revised: 7 May 2023
Accepted: 9 May 2023
Published: 10 May 2023

1. Introduction

The task of automatic text summarization is to generate a concise and informative summary of a text document. The rapid growth of digital content in recent years has made text summarization an increasingly important task. According to the way of summary generation, text summarization can be divided into two types: extractive and abstractive. Most abstracted text summarization models are based on the seq2seq framework [1] to produce a summary by sequentially generating words after encoding the entire document. In contrast, extractive models rely on identifying the most relevant information from the source material and presenting it in a condensed form. Abstractive models are capable of generating new content that is not present in the source material, allowing them to capture more complex ideas and convey them in a more natural and fluid way. However, abstractive models can also be more challenging to develop and require more advanced techniques, and they are more prone to errors and may not always capture the intended meaning of the original text [2]. Conversely, extractive models provide benefits in terms of factuality and efficiency [3].

In order to extract sentences that are suitable for summarization from a document, it is crucial to model the relationships among sentences [4]. In early stages, many models use recurrent neural networks (RNNs) [5,6] to capture these cross-sentence relationships. However, RNN-based models often struggle to capture long-distance dependencies at the sentence level, particularly when dealing with lengthy documents. The transformer model was proposed, and the attention mechanism used in it can help the model learn the relationships among sentences in a document, including long-range dependencies. Many transformer-based models have achieved good results in extractive summarization tasks, such as BERTSum [7], Pegasus [8], and T5 [9]. Despite their success, transformer-based models tend to have a large number of parameters and a complex structure, which leads to longer training times.

A more intuitive approach is to use a graph structure to model the relationships among sentences. Graph neural networks (GNNs) [10] have recently gained popularity in exploring cross-sentence relationships for text summarization tasks. GNNs are a type of deep learning model that represent data in the form of a graph structure, with nodes and edges capturing the features and relationships among data points. In the context of text summarization, GNNs have the potential to analyze the relationships among sentences in a document and identify important connections, which could help to create a more coherent and informative summary.

Constructing a graph structure is a key issue in using graph neural networks to solve summarization tasks. Several recent studies have attempted to create summary graphs using text representations in linguistics, such as Approximate Discourse Graph (ADG) [11] and Abstract Meaning Representation (AMR) graphs [12]. The aim of these studies is to use these graphical representations to better represent and understand natural language texts in order to improve text summarization tasks. However, the construction of these graphs often relies on external tools, which may lead to the accumulation of errors in the resulting graphs.

This study employs a multi-granularity heterogeneous graph to model documents, which goes beyond traditional approaches that only create graphs using sentence-level nodes. We include additional nodes in the graph to enhance the relationships among sentences, adding more semantic units that act as intermediaries that connect sentences. Each additional node represents a specific relationship among the sentences it connects. In contrast to traditional graph construction methods that only model sentence nodes, this paper models several other semantic nodes with different granularity, including words and topics. At the same time, the method considers the relationship among them, including co-occurrence relationships and topic distribution relationships, so as to construct the edges between nodes. The two types of nodes, namely, word nodes and topic nodes, serve as intermediaries for constructing meta-paths that contain sentence nodes, such as Sentence–Word–Sentence (SWS) and Sentence–Topic–Sentence (STS). We use the modified GAT [13] network to learn semantic information from two meta-paths to update the feature representation of sentence nodes. Finally, the feature representation of the sentences is used for binary classification to obtain the final sentences that should be included in the summary. The heterogeneous graph we constructed has the following advantages: Word nodes and topic nodes contain different levels of semantic information, which can be aggregated using GAT to enhance sentence representations. Furthermore, serving as intermediate nodes, word nodes and topic nodes can connect sentences that are distant from each other, thereby enriching cross-sentence relations.

The contributions of this work are summarized as follows:

- We designed a multi-granularity heterogeneous graph consisting of three node types: words, sentences, and topics;
- We propose an automatic text summarization model based on GAT that captures semantic information from nodes with different granularity. This allows us to embed sentence representations with varying levels of semantic information;
- We conducted experiments on two English news datasets, CNN/DM and NYT, and the experimental results outperformed the baseline models.

The paper is organized as follows: Section 2 presents the related work. In Section 3, our method is introduced in detail. Section 4 covers information on experiments, including the selected datasets, experimental hyperparameter settings, training objectives, and the baseline models and evaluation metrics. Experimental results on two datasets and detailed analysis are provided in Section 5. Finally, Section 6 provides the conclusion of the work presented in this paper, as well as prospects for future work.

2. Related Work

2.1. Extractive Document Summarization

Extractive document summarization is an important task in natural language processing (NLP). Many traditional approaches are based on statistical methods, such as TF-IDF [14,15] and graph-based ranking algorithms [16,17]. With the recent advances in deep learning, neural network-based methods have become the dominant approach to extractive summarization. Early neural network-based models focused on using recurrent neural networks (RNNs) [18–20] and convolutional neural networks (CNNs) [21,22] to model sentence representations. More recently, attention-based models such as transformer [23] architectures have been applied to capture the contextual dependencies among sentences. Additionally, some studies have explored the use of graph neural networks to model the relationships among sentences [24,25] and the use of topic modeling to enhance the summarization performance [26,27]. Overall, the development of more effective and efficient methods for extractive document summarization remains an active research area.

2.2. Graph Neural Network for Document Summarization

In recent years, graph neural networks (GNNs) [10] have been widely used in many NLP tasks, including text classification, relation extraction, question answering, and document summarization. Recently, there has been growing interest in using GNNs for document summarization. GNNs are a class of neural networks capable of handling structured data, such as graphs, and have been successfully applied to various tasks, such as node classification and graph classification [28,29]. In the task of text summarization, GNNs are able to model the relationships among sentences in a document and generate a summary by selecting the most relevant sentences [30].

Several studies have proposed GNN-based models for document summarization. For example, the DiscoBert model [31] combines the Rhetorical Structure Theory (RST) [32] graph and graph convolutional network (GCN) [33] to extract important sentences and learn the relationships among them. Another model, called GSum [34], utilizes a graph-based attention mechanism to capture the contextual information of sentences and generate a summary. In addition, some studies have explored the use of hierarchical GNNs to model the relationships between paragraphs and sentences in a document [35].

Overall, GNN-based models have shown promising results in document summarization and have the potential to overcome some of the limitations of traditional extractive approaches. To build a successful model for document summarization using GNNs, it is crucial to address the challenge of efficiently modeling documents as graphs. To tackle this challenge, this paper proposes a novel heterogeneous graph data structure.

3. Method

Extractive text summarization can be viewed as a sentence classification task, where the key is to identify the sentences that best capture the essential information of the original text. In this study, a heterogeneous graph structure was designed to represent the relationships among sentences, words, and topics. We use the neural topic model (NTM) [36] to obtain an initial representation of the topic nodes. The initial representation of words and sentences is obtained from the text vectorization layer. Additionally, GAT is used to aggregate word and implicit topic information in the graph structure data, enabling the sentence feature representation to contain richer and more accurate semantic information. Finally, the sentence feature representation is inputted into a sentence classifier to extract the summary sentences.

The architecture of our model is represented in Figure 1.

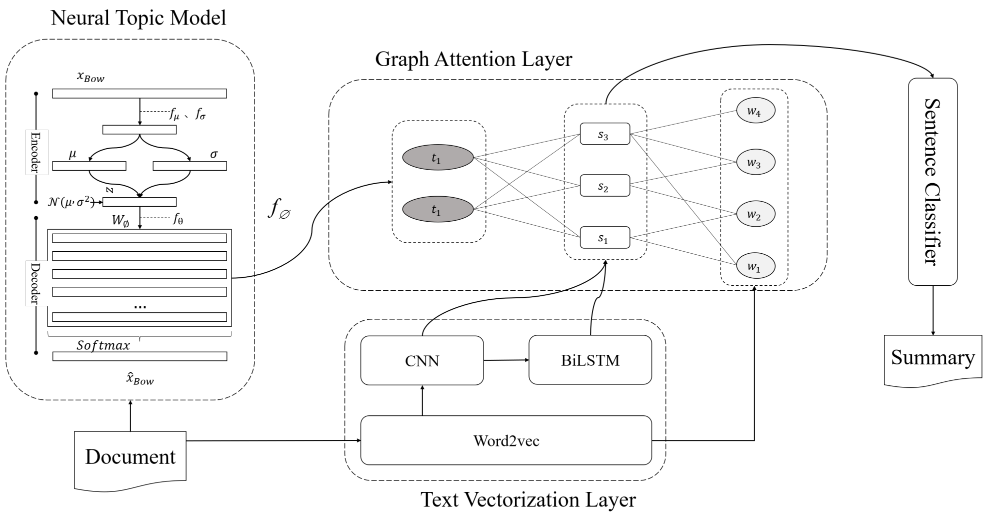


Figure 1. Overall architecture of the model. The framework consists of neural topic model, text vectorization layer, graph attention layer, and sentence classifier.

3.1. Text Vectorization

To obtain initial vectors for both words and sentences, we use GloVe [37], a pre-trained word embedding model, to obtain word vectors that capture the semantic and syntactic relationships among words. To obtain sentence vectors, the method first uses a CNN [38] to extract local feature vectors from word vectors. Let l_i denote the local feature vector of sentence i obtained by the CNN. By utilizing the local feature vectors extracted from the word vectors as input, a Bidirectional Long Short-Term Memory (BiLSTM) [39] layer is employed to process the sentence backward and forward, and ultimately capture a contextual representation g_i of the sentence. Finally, the local feature vectors and contextual representation are combined to obtain the sentence's initial feature vector, $s_i = [l_i; g_i]$, which represents the overall meaning of the sentence, taking into account both the local and global contexts.

3.2. Neural Topic Model

Inspired by Cui et al. [27], we add topic nodes to the heterogeneous graph structure. Specifically, we use the NTM to generate a feature representation of the topic nodes. The NTM uses an encoding–decoding process to learn the latent topics in the document. In the encoding phase, μ and σ are obtained based on word packet $x_{\text{bow}} \in \mathbb{R}^{|V|}$ for a given document, where $|V|$ is the size of the vocabulary list, and μ and σ are the average and variance of the potential issue distribution, i.e., $\mu = f_\mu(x_{\text{bow}})$ and $\log \sigma^2 = f_\sigma(x_{\text{bow}})$, respectively. Functions f_μ and f_σ are linear transformations with *ReLU* activation functions.

During the decoding phase, the NTM approximates the topic distribution with a Gaussian distribution using the previously calculated prior parameters μ and σ. This is achieved by generating a sample z from a normal distribution with mean μ and variance σ^2, i.e., $z \sim \mathcal{N}(\mu, \sigma^2)$. Afterward, we apply the softmax function to z to obtain a topic distribution $\theta \in \mathbb{R}^K$, where K denotes the number of topics.

To predict the probability distribution of the words, we multiply topic distribution θ by word distribution matrix $W_\varnothing$ and apply the softmax function. The resulting probability vector, $p_w \in \mathbb{R}^{|V|}$, represents the likelihood of the individual words in the vocabulary, where $|V|$ is the size of the vocabulary. Note that $W_\varnothing$ is equivalent to the topic–word distribution matrix in the Latent Dirichlet Allocation model.

Finally, we extract the words from the bag of words using their respective probability, p_w, and reconstruct the original text, χ_{bow}. For more detailed information about the NTM, readers are referred to the work by Miao et al. [36]. As intermediate parameters $W_\varnothing$ contain topical information, we utilize it to construct topic representations using the following approach:

$$H_T = f_{\varnothing}\left(W_{\varnothing}^T\right), \tag{1}$$

We use $H_T \in \mathbb{R}^{K \times d_t}$ to denote K topic representations, and each representation has a predetermined dimension of d_t. $f_{\varnothing}$ is a linear transformation activated with the ReLU function.

3.3. Graph Attention Layer

3.3.1. Graph Building

The graph structure data can be represented by $G = (V, E)$, where V denotes the set of nodes and E denotes the set of edges. Node set V in the heterogeneous graph constructed in this study consists of three parts, as shown in Figure 2: a word set $\{w_1, w_2, \ldots, w_m\}$, containing M word nodes; a sentence set $\{s_1, s_2, \ldots, s_n\}$, containing N sentence nodes; and a topic set $\{t_1, t_2, \ldots, t_k\}$, containing K topic nodes. In other words, $V = V_w \cup V_s \cup V_t$.

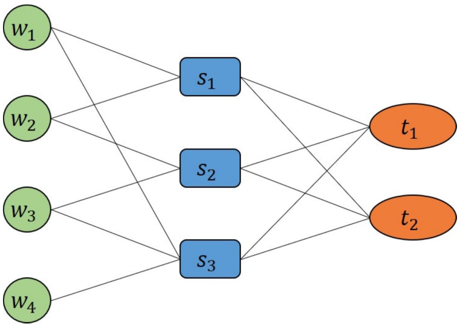


Figure 2. Heterogeneous graph schematic, where green indicates word nodes, blue indicates sentence nodes, and orange indicates topic nodes.

The set of edges, E, consists of two parts, of which one is E_{ws}, the set of edges representing word–sentence relations, and the other is E_{st}, the set of edges representing sentence–topic relations. The containment relationship between a sentence and a word is expressed as the existence of an edge. For each sentence node and topic node, the topic representation obtained using the NTM is used to calculate the edge weights, which indicate the correlations between sentences and topics.

3.3.2. Graph Propagation

We use GAT [13] and combine it with multi-headed attention mechanisms to update the node representation. To handle nodes with different feature dimensions, we use a learnable linear transformation matrix for each node to project its feature vector onto a common embedding space, where all feature vectors have the same dimension. For a sentence node with a feature vector $h_s \in \mathbb{R}^{d_s}$, we learn a weight matrix $W_s \in \mathbb{R}^{d_h \times d_s}$ to transform it into a new feature vector, $h \in \mathbb{R}^{d_h}$, where d_h is the desired dimension of the embedding space. We apply the same strategy to obtain new feature vectors, h_w and h_t, for word nodes and topic nodes, respectively. Then, for each node i, we compute attention coefficients α_{ji} between node i and its neighbors using the following equation:

$$z_{ij} = \text{LeakyReLU}\left(a^T * \left[W_q h_i \| W_k h_j\right]\right), \tag{2}$$

$$\alpha_{ij} = \frac{\exp\left(z_{ij}\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(z_{ik}\right)}, \tag{3}$$

where $\|$ denotes concatenation; $\mathcal{N}_i$ is the set of neighbors of node i; and a, W_q, and W_k are learnable weights.

Next, we compute the attention output for each node i by aggregating the features of its neighbors using the attention coefficients, as follows:

$$u_i = \sum_{j \in \mathcal{N}_i} a_{ij} W_v h_j, \tag{4}$$

where W_v are learnable weights. Then, we concatenate the outputs of multiple attention heads and apply a final linear projection to obtain the final representation of node i, as follows:

$$h_i' = \sigma\left(\frac{1}{M} \sum_{M=1}^{M} W_m * u_i^m\right), \tag{5}$$

where M is the number of attention heads, u_i^m is the output of the m-th attention head for node i, and W_m is the learnable weight matrix for the m-th head. The model is trained using a cross-entropy loss function, and a back-propagation algorithm is used to update the model parameters. Finally, the node representation can be used for classification tasks.

3.4. Sentence Selector

We use a fully connected layer followed by a sigmoid activation function. The fully connected layer [40] projects the node representations onto a lower dimensional space, and the sigmoid function squashes the output to the range [0, 1], representing the probability of the node belonging to one of the two classes.

Formally, given the final node representations, h, the output of the binary classification task can be computed as

$$y = \sigma\left(W_f h + b_f\right) \tag{6}$$

where W_f and b_f are the weight matrix and the bias term of the fully connected layer, respectively, and σ is the sigmoid activation function. Output y is a scalar value between 0 and 1, representing the probability of the node belonging to the positive class. The binary label of the node can then be determined based on a threshold value.

4. Experimental Setup

4.1. Datasets

CNN/DM: The CNN/DailyMail dataset [41] is a widely used document summarization benchmark dataset that includes news articles collected from the CNN and DailyMail websites, as well as human-written summaries. The training set contains approximately 287,000 articles and their corresponding summaries, while the validation and test sets each contain approximately 13,000 articles and summaries. These articles are relatively long, with an average length of about 800 words. We preprocessed the data following the method by Liu and Lapata et al. [42].

NYT50: NYT50 is a dataset [43] for document summarization and consists of 50,000 news articles from the New York Times. Each article has a headline and a one-sentence summary that serves as a base-truth summary. The dataset covers a diverse range of topics and is a challenging benchmark for document summarization. We followed the approach proposed by Durrett et al. [43], who used the last 4000 examples from the training set for validation and selected 3452 examples as the test set by filtering out those that did not meet certain criteria.

4.2. Training Objective

Extractive text summarization can be performed as a binary classification task on sentence nodes, and the loss function can be expressed as

$$\mathcal{L}_{SC} = \sum_{i=1}^{n} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)), \tag{7}$$

where n is the number of samples, y_i is the true label of the i-th sample, and $\hat{y}_i$ is the label predicted by the model. The loss function of the NTM model in this paper can be described as

$$\mathcal{L}_{NTM} = D_{KL}(p(z) \| q(z \mid x)) - \mathbb{E}_{q(\varepsilon|x)}[p(x \mid z)], \tag{8}$$

where $D_{KL}(p(z) \| q(z \mid x))$ is the Kullback–Leibler divergence loss between $p(z)$ and $q(z \mid x)$, which measures the loss of information from prior distribution $p(z)$ to posterior distribution $q(z \mid x)$, and $\mathbb{E}_{q(\varepsilon|x)}[p(x \mid z)]$ is the reconstruction error term, which represents the error of sampling a topic z from posterior distribution $q(z \mid x)$ and then reconstructing input x using generative model $p(x \mid x)$.

We weight the loss function of the NTM and the cross-entropy loss function together to obtain a total loss function.

$$\mathcal{L} = \mathcal{L}_{SC} + \alpha \mathcal{L}_{NTM}, \tag{9}$$

where α is the preset hyperparameter.

4.3. Settings and Hyperparameters

In our approach, we first filtered out stop words and punctuation while creating the word nodes, as these words typically do not contribute much to the meaning of a sentence. In addition, we wanted to eliminate noisy common words, so we further removed 20% of the vocabulary with low $TF - IDF$ values across the entire dataset. By doing so, we were able to focus on the most important words and phrases that captured the essence of the text.

To encode the documents, we generated 300-dimensional word vectors using GloVe embedding and initialized sentence nodes with a size of $ds = 128$.

For the NTM model, we set the number of topics to $K = 50$ and the dimension size of the topic representation to 512.

For the GAT model, we implemented it using Pytorch Geometric [44], a popular library for building graph neural networks. We set the number of GAT layers to two, the number of attention heads to six, and the size of the hidden layers to 128.

During the training phase, we used Adam Optimizer [45] at a learning rate of 5×10^{-4} and a batch size of 16. Since the convergence rate of the NTM model is slower than that of the GAT model, we pre-trained the NTM model for 250 epochs at a learning rate of 1×10^{-3}, allowing it to learn a good initial set of parameters before fine-tuning on the task of summarization. Moreover, as a criterion during the sentence selection process for the CNN/DailyMail and NYT50 datasets, we based our selection of the top-three sentences on the average length of the corresponding human-written summaries.

4.4. Baseline Models and Evaluation Metrics

We selected several extractive summarization models that have been proposed in recent years as the baseline models for comparison.

LEAD-3 is a simple text summarization algorithm that uses the first few sentences of the text to generate a summary.

Oracle is a theoretical upper bound model to measure the performance of our model.

NeuSum [19] is based on the seq2seq framework, which uses an attention mechanism to focus on the most relevant parts of the input text when generating summaries.

JECS [46] is a deep learning-based summarization model that can both extract and compress the key information in the original text.

BERTSUM [42] is a text summarization method based on the pre-trained language model BERT.

PNBERT [47] is also an effective neural extractive summarization algorithm based on the pre-trained model BERT.

HSG [4] is a heterogeneous graph-based text summarization model that uses a heterogeneous graph containing word and sentence nodes.

HiBERT [48] is a pre-trained language model specifically designed for document-level language understanding tasks.

To evaluate the quality of the generated summaries, we used three commonly used evaluation metrics in the field of summarization: ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L). ROUGE [49] stands for Recall-Oriented Understudy for Gisting Evaluation, and it measures the overlap between the generated summary and the reference summary in terms of n-gram recall.

5. Results and Analysis

5.1. Main Results

Table 1 presents the experimental results, which are divided into four sections. The first section displays the ROUGE scores of the LEAD-3 and Oracle models. The second section contains several extractive models that do not use the pre-trained language model BERT. The third section contains several extractive models that use the pre-trained language model BERT. The final section reports the results of our model.

Table 1. Comparison of model performance. A dash (-) indicates that the corresponding result was not reported. The bolded values indicate the highest value in the current column.

Model	CNN/DM			NYT		
	R-1	R-2	R-L	R-1	R-2	R-L
LEAD-3	40.42	17.62	36.67	41.80	22.60	35.00
Oracle	55.61	32.84	51.88	64.22	44.57	57.27
NeuSum [19]	41.59	19.01	37.98	-	-	-
JECS [46]	41.70	18.50	37.90	45.50	25.30	39.20
HSG [4]	42.31	19.51	38.74	46.89	26.26	42.58
BERTSUM [42]	**43.25**	**20.24**	**39.63**	-	-	-
PNBERT [47]	42.69	19.60	38.85	-	-	-
HiBERT [48]	42.37	19.95	38.83	**49.06**	**29.70**	41.23
MGH-Sum (our model)	42.96	19.81	39.12	47.88	28.68	**42.65**

Observing the experimental results, we draw the following conclusions: In terms of Rouge-1, Rouge-2, and Rouge-L scores, our model outperformed all models that do not use a pre-trained language model and performed competitively compared with models that use a pre-trained language model. Specifically, our model achieved higher ROUGE-1, ROUGE-2, and ROUGE-L scores, 42.96, 19.81, and 39.12, respectively, on the CNN/DM dataset than the best-performing model that does not use a pre-trained language model, HSG. Furthermore, our model also outperformed the PNBERT and HiBERT models in terms of ROUGE-1 and ROUGE-L scores, while on the NYT dataset, our model achieved ROUGE-1, ROUGE-2, and ROUGE-L scores of 47.88, 28.68, and 42.65, respectively, outperforming all models that do not use a pre-trained language model.

These results indicate that our proposed model is effective in capturing important information and relationships among sentences, words, and topics and in generating summaries that contain key information from the original texts. The performance of our model suggests that heterogeneous graph neural networks are a promising approach to extractive text summarization, particularly in domains where pre-training data may be limited or where pre-trained language models may not be optimal.

5.2. Ablation Study

In our study, we conducted an ablation study to analyze the effectiveness of each component in our proposed model. By selectively removing or modifying specific parts of the model, we are able to evaluate the impact of each component on the model's performance on a specific task and identify which parts were most critical to achieving high performance. This enabled us to gain a deeper understanding of our model's functionality and behavior while also offering valuable insights for future improvements.

Specifically, we set up three ablation models to conduct our experiments:

- w/o GAT, which refers to the removal of the part that uses GAT in heterogeneous graph neural networks, i.e., not using GAT for feature aggregation and information propagation between nodes;
- w/o word nodes, which refers to the removal of the part that considers word nodes in heterogeneous graph neural networks, i.e., not taking into account the influence of word nodes and only using the features of other types of nodes for graph structure information aggregation and propagation;
- w/o topic nodes, which, similarly to the "w/o word nodes" model, refers to the removal of the part that uses topic nodes in heterogeneous graph neural networks.

Figure 3 shows the experimental results of different variants of our model on the CNN/DM and NYT datasets. Based on the results of the ablation experiments, we draw the following conclusions:

1. Our complete model achieved the best performance, which shows that the integration of all its components is essential to achieving the best results;
2. The model's performance was most significantly reduced when the GAT layer was removed, possibly because without the GAT layer, the node representations cannot be updated. In this case, the model becomes similar to the EXT-BiLSTM model [4];
3. On different datasets, the gains of word nodes and topic nodes were different. Specifically, on the CNN/DM dataset, removing the word nodes led to more significant performance degradation than removing the topic nodes. However, on the NYT dataset, retaining topic nodes led to better performance in the ROUGE-1 and ROUGE-2 scores. The reason for this phenomenon may be due to the different styles of articles in the two datasets: articles in the CNN/DM dataset are usually more concise and clear, focusing on factual statements and objective reporting, while articles in the NYT dataset focus more on in-depth reporting and critical analysis. For these reasons, we speculate that incorporating implicit topic information is more advantageous for generating complex article summarization.

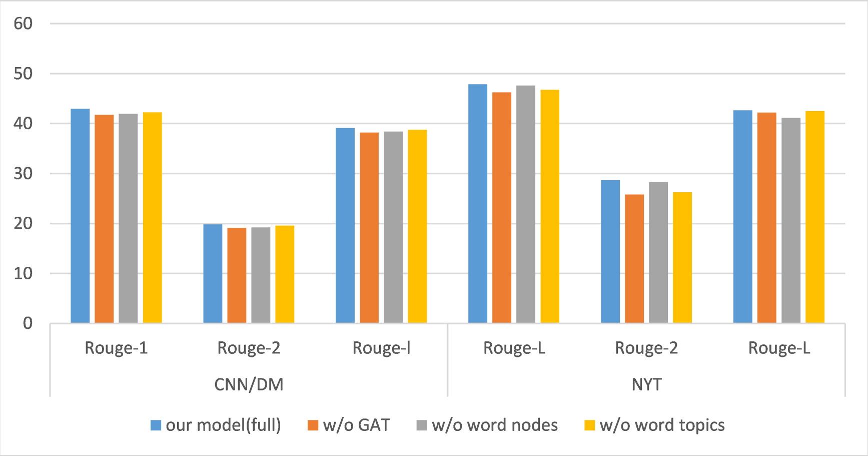


Figure 3. Effect of model components on evaluation metrics. The figure shows the performance metrics of the full model and the three ablation models on the CNN/DM and NYT datasets.

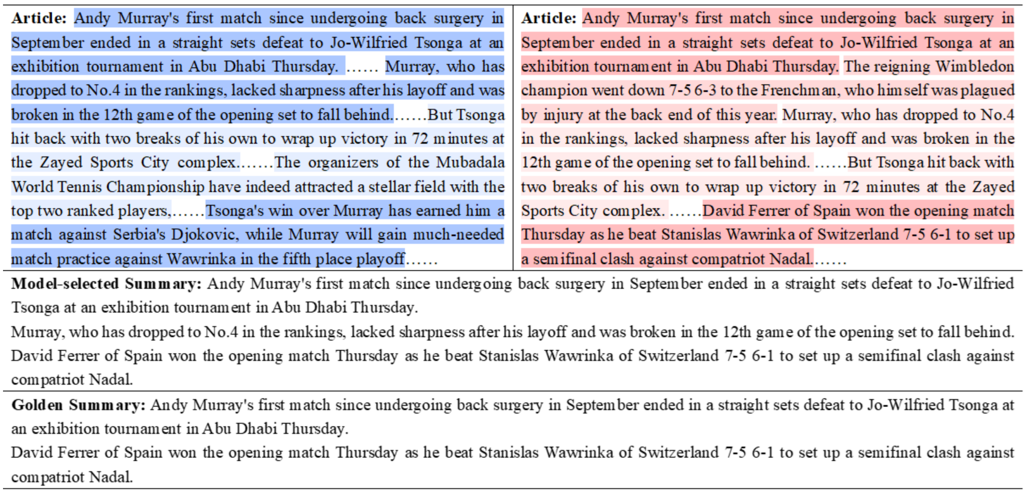
5.3. Qualitative Analysis

In this subsection, we present the experiments conducted to demonstrate how the introduction of word and topic nodes in a heterogeneous graph enhances the model's ability to accurately summarize documents. We employed an attention mechanism to compute the mutual interactions among nodes in our heterogeneous graph. To conduct a qualitative analysis of a specific node type k, we propose calculating the attention weights that are summed between each node v and all the nodes belonging to k. These weights can be computed using the following formula:

$$w_i^k = \sum_{j \in T_k} \alpha^k{}_{i,j} \tag{10}$$

where T_k represents the set of nodes belonging to type k, $\alpha^k{}_{i,j}$ (Equation (3)) denotes the attention weight between node i and all nodes j that belong to type k, and w_i^k represents the influence of nodes belonging to type k on target node i.

According to Equation (10), the sum of attention weights from word and topic nodes can be separately calculated for each sentence in a document. We denote the attention weights from the word nodes as word-level attention weights and the attention weights from the topic nodes as topic-level attention weights. An example of the visualization of attention weights from word and topic nodes for each sentence in a CNN/DM article is depicted in Figure 4. After examining this example, we found that the ground-truth summary sentences often obtained high scores in both word-level and topic-level attention weights. Furthermore, the sentences selected by our model exhibited a high degree of overlap with these ground-truth sentences. With this observation, we can develop an intuitive understanding of how our model works: The graph attention layer establishes connections among words, sentences, and topics, and the GAT model integrates both word-level and topic-level attention weights while considering the local features and contextual relationships among sentences, ultimately resulting in the accurate selection of summary sentences. This strategic approach guides our model's attention to focus on specific sentences that may carry crucial information, rather than requiring it to analyze the entire document.



Article: Andy Murray's first match since undergoing back surgery in September ended in a straight sets defeat to Jo-Wilfried Tsonga at an exhibition tournament in Abu Dhabi Thursday. …… Murray, who has dropped to No.4 in the rankings, lacked sharpness after his layoff and was broken in the 12th game of the opening set to fall behind.……But Tsonga hit back with two breaks of his own to wrap up victory in 72 minutes at the Zayed Sports City complex.……The organizers of the Mubadala World Tennis Championship have indeed attracted a stellar field with the top two ranked players,……Tsonga's win over Murray has earned him a match against Serbia's Djokovic, while Murray will gain much-needed match practice against Wawrinka in the fifth place playoff……

Article: Andy Murray's first match since undergoing back surgery in September ended in a straight sets defeat to Jo-Wilfried Tsonga at an exhibition tournament in Abu Dhabi Thursday. The reigning Wimbledon champion went down 7-5 6-3 to the Frenchman, who himself was plagued by injury at the back end of this year. Murray, who has dropped to No.4 in the rankings, lacked sharpness after his layoff and was broken in the 12th game of the opening set to fall behind. ……But Tsonga hit back with two breaks of his own to wrap up victory in 72 minutes at the Zayed Sports City complex. ……David Ferrer of Spain won the opening match Thursday as he beat Stanislas Wawrinka of Switzerland 7-5 6-1 to set up a semifinal clash against compatriot Nadal.……

Model-selected Summary: Andy Murray's first match since undergoing back surgery in September ended in a straight sets defeat to Jo-Wilfried Tsonga at an exhibition tournament in Abu Dhabi Thursday.
Murray, who has dropped to No.4 in the rankings, lacked sharpness after his layoff and was broken in the 12th game of the opening set to fall behind.
David Ferrer of Spain won the opening match Thursday as he beat Stanislas Wawrinka of Switzerland 7-5 6-1 to set up a semifinal clash against compatriot Nadal.

Golden Summary: Andy Murray's first match since undergoing back surgery in September ended in a straight sets defeat to Jo-Wilfried Tsonga at an exhibition tournament in Abu Dhabi Thursday.
David Ferrer of Spain won the opening match Thursday as he beat Stanislas Wawrinka of Switzerland 7-5 6-1 to set up a semifinal clash against compatriot Nadal.

Figure 4. Visualization of word-level attention weights and topic-level attention weights for each sentence in a CNN/DM article. The left side shows word-level attention weights and the right side shows topic-level attention weights. The degree of highlighting represents a greater impact or influence on the sentence from either word or topic nodes.

6. Conclusions

In this paper, we investigated the impact of introducing more nodes into a heterogeneous graph on document summarization. We propose a new summarization graph structure that includes not only sentence nodes but also additional word and topic nodes. To help the model extract more accurate summary sentences containing important information, we utilize the GAT network to update the nodes in the graph, aggregating semantic information from both word-level and topic-level nodes and utilize this information to enrich the representation of sentences. Experimental results on the CNN/DM and NYT datasets show that our model outperformed the baseline models that did not use pre-trained language models. In future work, we plan to enhance the performance in text summarization by introducing various semantic nodes, such as entities, into the summary graph.

Author Contributions: Conceptualization, H.Z. and W.Z.; methodology, H.Z.; software, H.Z.; validation, H.Z.; formal analysis, H.Z. and W.Z.; investigation, H.Z.; resources, W.Z.; data curation, Y.W. and S.F.; writing—original draft preparation, H.Z.; writing—review and editing, H.Z. and W.Z.; visualization, H.Z.; supervision, W.Z., Y.W. and S.F.; project administration, W.Z., M.H., S.F. and Y.W.; funding acquisition, W.Z. and M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by The National Natural Science Foundation of China (Nos. 82260362 and 62241202) and in part by The National Key R&D Program of China (No. 2021ZD0111000).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *arXiv* **2014**, arXiv:1409.3215.
2. Liu, Y.; Lapata, M. Hierarchical transformers for multi-document summarization. *arXiv* **2019**, arXiv:1905.13164.
3. Cao, Z.; Wei, F.; Li, W.; Li, S. Faithful to the original: Fact aware neural abstractive summarization. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
4. Wang, D.; Liu, P.; Zheng, Y.; Qiu, X.; Huang, X. Heterogeneous graph neural networks for extractive document summarization. *arXiv* **2020**, arXiv:2004.12393.
5. Cheng, J.; Lapata, M. Neural summarization by extracting sentences and words. *arXiv* **2016**, arXiv:1603.07252.
6. Nallapati, R.; Zhai, F.; Zhou, B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
7. Liu, Y. Fine-tune BERT for extractive summarization. *arXiv* **2019**, arXiv:1903.10318.
8. Ahmed, R.; DeSmedt, P.; Du, W.; Kent, W.; Ketabchi, M.A.; Litwin, W.A.; Rafii, A.; Shan, M.C. The Pegasus heterogeneous multidatabase system. *Computer* **1991**, *24*, 19–27. [CrossRef]
9. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
10. Duvenaud, D.K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R.P. Convolutional networks on graphs for learning molecular fingerprints. *arXiv* **2015**, arXiv:1509.09292.
11. Yasunaga, M.; Zhang, R.; Meelu, K.; Pareek, A.; Srinivasan, K.; Radev, D. Graph-based neural multi-document summarization. *arXiv* **2017**, arXiv:1706.06681.
12. Wang, T.; Wan, X.; Jin, H. Amr-to-text generation with graph transformer. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 19–33. [CrossRef]
13. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
14. Savyanavar, P.; Mehta, B.; Marathe, V.; Padvi, P.; Shewale, M. Multi-document summarization using TF-IDF Algorithm. *Int. J. Eng. Comput. Sci.* **2016**, *5*, 16253–16256. [CrossRef]
15. Christian, H.; Agus, M.P.; Suhartono, D. Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). *ComTech Comput. Math. Eng. Appl.* **2016**, *7*, 285–294. [CrossRef]
16. Mihalcea, R.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
17. Erkan, G.; Radev, D.R. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.* **2004**, *22*, 457–479. [CrossRef]
18. Pan, S.; Li, Z.; Dai, J. An improved TextRank keywords extraction algorithm. In Proceedings of the ACM Turing Celebration Conference-China, Chengdu, China, 17–19 May 2019; pp. 1–7.
19. Zhou, Q.; Yang, N.; Wei, F.; Huang, S.; Zhou, M.; Zhao, T. Neural document summarization by jointly learning to score and select sentences. *arXiv* **2018**, arXiv:1807.02305.
20. Nallapati, R.; Zhou, B.; Gulcehre, C.; Xiang, B. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv* **2016**, arXiv:1602.06023.
21. Tan, J.; Wan, X.; Xiao, J. Abstractive document summarization with a graph-based attentional neural model. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1171–1181.

22. Chen, Y. Convolutional Neural Network for Sentence Classification. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2015.
23. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:1404.2188.
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
25. Liu, Y.; Safavi, T.; Dighe, A.; Koutra, D. Graph summarization methods and applications: A survey. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–34. [CrossRef]
26. AL-Khassawneh, Y.A.; Hanandeh, E.S. Extractive Arabic Text Summarization-Graph-Based Approach. *Electronics* **2023**, *12*, 437. [CrossRef]
27. Cui, P.; Hu, L.; Liu, Y. Enhancing extractive text summarization with topic-aware graph neural networks. *arXiv* **2020**, arXiv:2010.06253.
28. Gu, Y.; Wang, Y.; Zhang, H.R.; Wu, J.; Gu, X. Enhancing Text Classification by Graph Neural Networks With Multi-Granular Topic-Aware Graph. *IEEE Access* **2023**, *11*, 20169–20183. [CrossRef]
29. Zhang, H.; Lu, G.; Zhan, M.; Zhang, B. Semi-supervised classification of graph convolutional networks with Laplacian rank constraints. *Neural Process. Lett.* **2021**, *54*, 2645–2656. [CrossRef]
30. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef]
31. Xu, J.; Gan, Z.; Cheng, Y.; Liu, J. Discourse-aware neural extractive text summarization. *arXiv* **2019**, arXiv:1910.14142.
32. Mann, W.C.; Thompson, S.A. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdiscip. J. Study Discourse* **1988**, *8*, 243–281. [CrossRef]
33. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
34. Dou, Z.Y.; Liu, P.; Hayashi, H.; Jiang, Z.; Neubig, G. Gsum: A general framework for guided neural abstractive summarization. *arXiv* **2020**, arXiv:2010.08014.
35. Jia, R.; Cao, Y.; Tang, H.; Fang, F.; Cao, C.; Wang, S. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 3622–3631.
36. Miao, Y.; Grefenstette, E.; Blunsom, P. Discovering discrete latent topics with neural variational inference. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 2410–2419.
37. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
38. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
39. Graves, A.; Graves, A. Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45.
40. McClelland, J.L.; Rumelhart, D.E.; PDP Research Group. *Parallel Distributed Processing*; MIT Press: Cambridge, MA, USA, 1986; Volume 2.
41. Hermann, K.M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching machines to read and comprehend. *arXiv* **2015**, arXiv:1506.03340.
42. Liu, Y.; Lapata, M. Text summarization with pretrained encoders. *arXiv* **2019**, arXiv:1908.08345.
43. Durrett, G.; Berg-Kirkpatrick, T.; Klein, D. Learning-based single-document summarization with compression and anaphoricity constraints. *arXiv* **2016**, arXiv:1603.08887.
44. Fey, M.; Lenssen, J. Fast graph representation learning with PyTorch Geometric. *arXiv* **2019**, arXiv:1903.02428.
45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
46. Xu, J.; Durrett, G. Neural extractive text summarization with syntactic compression. *arXiv* **2019**, arXiv:1902.00863.
47. Zhong, M.; Liu, P.; Wang, D.; Qiu, X.; Huang, X. Searching for effective neural extractive summarization: What works and what's next. *arXiv* **2019**, arXiv:1907.03491.
48. Zhang, X.; Wei, F.; Zhou, M. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv* **2019**, arXiv:1905.06566.
49. Lin, C. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.