Multi Graph Neural Network for Extractive Long Document Summarization

Xuan-Dung Doan[1] and **Le-Minh Nguyen**[2] and **Khac-Hoai Nam Bui**[1]*
[1]Viettel Cyperspace Center, Viettel Group, Vietnam
[2]Japan Advanced Institute of Science and Technology, Japan
dungdx4@viettel.com.vn,nguyenml@jaist.ac.jp,nambkh@viettel.com.vn

Abstract

Heterogeneous Graph Neural Networks (HeterGNN) have been recently introduced as an emergent approach for extracting document summarization (EDS) by exploiting the cross-relations between words and sentences. However, applying HeterGNN for long documents is still an open research issue. One of the main majors is the lacking of inter-sentence connections. In this regard, this paper exploits how to apply HeterGNN for long documents by building a graph on sentence-level nodes (homogeneous graph) and combine with HeterGNN for capturing the semantic information in terms of both inter and intra-sentence connections. Experiments on two benchmark datasets of long documents such as PubMed and ArXiv show that our method is able to achieve state-of-the-art results in this research field.

1 Introduction

Extractive Document summarization aims to automatically extract a set of sentences, which represents information for the whole document, by ranking the importance of sentence features. Recent works focus on GNN, a Deep learning-based approach that operates on graph domain (Zhou et al., 2020), to achieve remarkable results in this research field. Specifically, GNN-based models are able to encode the complicated pairwise relationships between entity tokens for better informative representations (Wu et al., 2021). Cui et al. (2020) uses information of topic-aware to change the representation of words to a new representation. Then, a GNN-based model is presented for capturing relationships efficiently via graph-structured document representation between sentences. Sequentially, HeterGNN, a special kind of GNN (Zhang et al., 2019), has been proposed as a promising approach to enrich the relationships between words and sentences. Wang et al. (2020) introduced HSG, a heterogeneous graph-based neural network for extractive summarization by using more fine-grained semantic units in the summarization graph to extract the complex relationships between words and sentences. Accordingly, the model has achieved the top performance in CNN/DailyMail and NYT50 datasets in terms of the non-BERT-based approach. In order to utilize the capability of BERT-based models (Devlin et al., 2019), Jia et al. (2020) proposed a hierarchical attentive heterogeneous graph (HAHSum) to improve the redundant phrases problem between extracted sentences of the summarization, which has achieved promising results on news article datasets such as CNN/DailyMail, NYT, and Newsroom. Nevertheless, the model requires external analysis for modeling long-range dependencies. Intuitively, transformer-based language models are not able to process long pieces of text. Several works have provided promising results (Cui and Hu, 2021), however, the input length limitation and encoding of long texts are still open challenges in this research field (Zhong et al., 2020).

In this study, we take an investigation on improving the performance of the EDS problem for long documents in which the core idea is to exploit the complex relationship in terms of both inter and intra-sentence connections using graph-based methods. Specifically, HeterGNN-based models are able to enrich the cross-sentence relations by adding a word node as the intermediary to connect sentences. However, the inter-sentence connections are not considered. Specifically, only sentences with common words can have a connection, which might influence the performance, especially in terms of long-form document representation. Therefore, we present a novel method for enriching the inter-sentences relations by proposing a homogeneous graph neural network (HomoGNN) and incorporating the HeterGNN for final sentence representations. In particular, inspired by recent state-of-the-art models for long-form document represen-

*Corresponding author

Proceedings of the 29th International Conference on Computational Linguistics, pages 5870–5875
October 12–17, 2022.

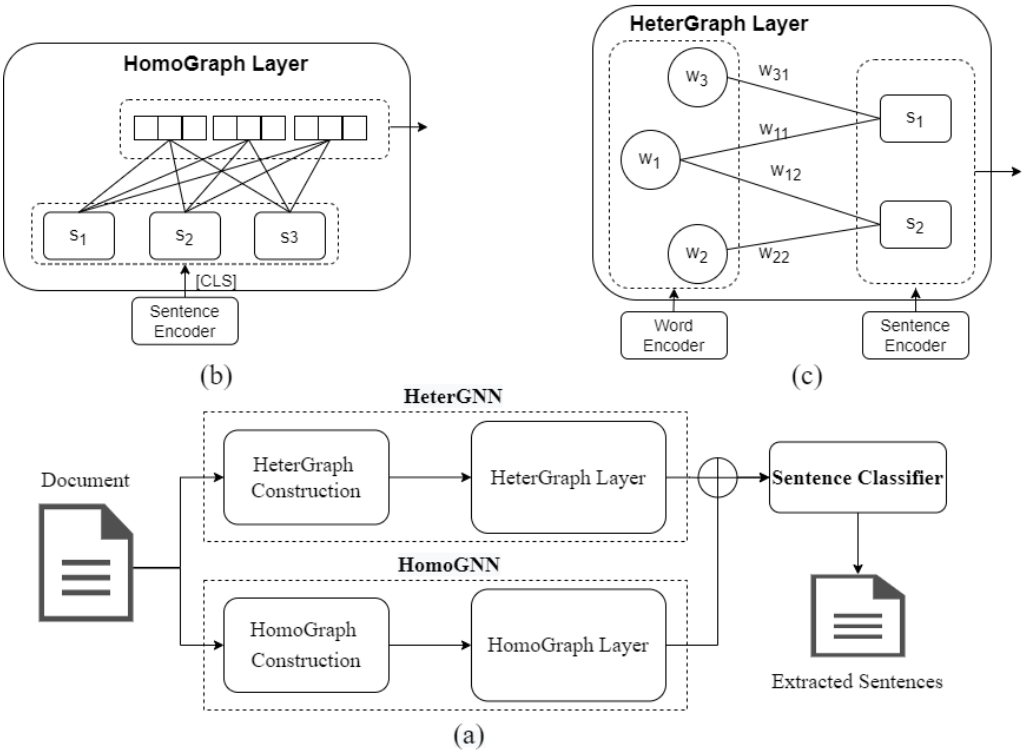

Figure 1: Overview pipeline of the proposed model which is executed simultaneously in two phases (a). The first phase encodes the sentences with pre-trained BERT and uses [CLS] information as the input of a graph attention layer (b). The second phase encodes the word and sentence nodes as the inputs of a heterogeneous graph layer (c). The output of the two phases is concatenated and put into an MLP layer in order to classify labels for each sentence.

tations such as Longformer (Beltagy et al., 2020), Big-Bird (Zaheer et al., 2020), and Poolingformer (Zhang et al., 2021), we use the information at the beginning of the sentence [CLS] representation for the inputs of the graph attention layer. Sequentially, the combination of HomoGNN and HeterGNN is able to capture the semantic information for both inter and intra-sentence connections. Figure 1 illustrates the overview of the proposed model. To the best of our knowledge, our method is the first study to incorporate two types of graph structures for the EDS task. Our source code is available for further investigation on Github[1].

2 Methodology

Given an arbitrary document $D = \{s_1, .., s_n\}$ consisting n sentences, the objective of EDS problem is to predict a sequence of a set of binary label $\{y_1, .., y_n\}$. Specifically, $y_j \in [0, 1]$ represents the j^{th} sentence, which should be included in the summary. Our proposed model for the EDS problem includes two learning layers, which execute simultaneously, such as the homogeneous graph layer and the heterogeneous graph layer.

2.1 Homogeneous Graph Neural Network

Graph Construction: Let $G_1 = \{V_1, E_1\}$ denotes an arbitrary graph, where V_1 and E_1 represent the set of node and edge, respectively. Consequentially, the homogeneous graph for an input document can be defined as a set of node $V_1 = s_1, ..., s_n$ where n is the number of sentence in the document. For initialized encoder process, BERT (Devlin et al., 2019) is used to generate the local hidden representations between sentences. Specifically, we adopt the concept of BERTSUM(Liu and Lapata, 2019) with multiple CLS for sentence representation. Sequentially, *CLS* and *SEP* tokens are inserted at the beginning and end of each sentence, respectively. Then, all tokens are fed into BERT to learn the hidden state, which can be denoted as follows:

$$h_{1,0}, h_{1,1}, ..., h_{n,0}, ..., h_{n,*} = BERT(w_{1,0}, w_{1,1}, ..., w_{n,0}, ..., w_{n,*}) \quad (1)$$

where $w_{i,j}$ is the vector embedding of the sentence i^{th} and word j^{th}. $w_{i,0}$ and $w_{i,*}$ represents the *CLS* and *SEP* tokens of the i^{th} sentence, respectively.

[1] https://github.com/dungdx34/MTGNN-SUM

$h_{i,j}$ stands for the hidden state of the corresponding token. After BERT encoding, we select the hidden state of *CLS* to represent sentence contextual representation, which is formulated as follows:

$$H_B = h_{1,0}, ..., h_{N,0} \tag{2}$$

Sequentially, the initialized embedding is put into a GAT model for enriching the sentence connections.

Graph Propagation: Regarding the message passing process, we adopt GAT model (Velickovic et al., 2018) to learn the hidden representation of each node by aggregating the information from its neighbors. Specifically, the updated node with GAT can be calculated as follows:

$$z_{ij} = LeakyReLU(W_a[W_q h_i; W_e h_j]) \tag{3}$$

where h_i represents the node representation of the i^{th} sentence. W_a, W_q, W_e, and W_v are trainable weights. Subsequently, the attention score between two sentence nodes is formulated as follows:

$$\begin{aligned} \alpha_{ij} = softmax(z_{ij}) = \frac{exp(z_{ij})}{\sum_{l \in N_i} exp(z_{il})} \\ \mu_i = \sigma(\sum_{j \in N_i} \alpha_{ij} W_v h_j) \end{aligned} \tag{4}$$

where σ denotes an activation function, and N_i stand for neighbor nodes. Consequentially, the output with multi-head attention can be calculated as follows:

$$h'_i = ||_{k=1}^{K} \sigma(\sum_{j \in N_i} \alpha_{ij}^k W_v^k h_j) \tag{5}$$

where $||*$ represents multi-heads concatenation. Furthermore, a residual connection is adopted to avoid gradient vanishing after iterations. Consequentially, the final output can be updated as follows:

$$H_s^{G_1} = h'_i + h_i \tag{6}$$

Generally, we use GAT for H_B to learn the relationship between sentences in a document. The output includes the representation of sentences, which is concatenated with the output of the heterogeneous graph layer for the final representation of the sentences.

2.2 Heterogeneous Graph Neural Network

Graph Construction: Let $G_2 = \{V_2, E_2\}$ denotes an undirected graph for representing the input document. The heterogeneous graph for an input document can be defined as $V_2 = V_w \cup V_s$ and $E_2 = \{e_{11}, ..., e_{mn}\}$, where $V_w = \{w_1, ..., w_m\}$ and $V_s = \{s_1, ..., s_n\}$ represents m unique words and n sentences of a document, respectively. e_{ij} denotes the edge between the *i*-th word and *j*-th sentence. Following the concept of HeterSumGraph (Wang et al., 2020), sentence node features are calculated by combining CNN for extracting the local n-gram feature of each sentence and bidirectional Long Short-Term Memory (BiLSTM) for extracting the sentence-level feature.

Graph Propagation: The heterogeneous graph layer is also updated using GAT, which is defined from Equation 3 to Equation 6. However, the vanilla GAT has been designed for homogeneous graphs. Therefore, Wang et al. (2020) has presented a modified GAT and an iterative updating mechanism for heterogeneous graph updated layer. Specifically, the Equation 3 can be re-formulated as follows:

$$z_{ij} = LeakyReLU(W_a[W_q h_i; W_e h_j; \overline{e}_{ij}]) \tag{7}$$

where $\overline{e}_{ij}$ denotes the multi-dimensional embedding space ($\overline{e}_{ij} \in \mathbb{R}^{mn \times d_e}$), which is mapped from edge weight e_{ij}. Sequentially, an iterative updating mechanism is adopted to obtain a new word node and sentence node. In particular, in order to pass messages between word and sentence nodes, the sentences with their neighbor word nodes are updated via modified-GAT and Position-Wise Feed-Forward (FFN) layer, which can be formulated as follows:

$$\begin{aligned} U_{s \leftarrow w}^1 = GAT(H_s^0, H_w^0, H_w^0) \\ H_s^1 = FFN(U_{s \leftarrow w}^1 + H_s^0) \end{aligned} \tag{8}$$

where H_w^0 (H_w^1) and H_s^0 are the node features of word X_w ($X_w \in \mathbb{R}^{m \times d_w}$) and sentence X_s ($X_s \in \mathbb{R}^{n \times d_s}$), respectively. Note that H_s^0 is used as the attention query and H_w^0 is regarded as key and value. Sequentially, the new representations of word nodes can be obtained using the updated sentence nodes and further updated sentence or query nodes, iteratively. Specifically, each iteration contains a sentence-to-word and a word-to-sentence update process, which is formulated as follows:

$$\begin{aligned} U_{w \leftarrow s}^{t+1} = GAT(H_w^t, H_s^t, H_s^t) \\ H_w^{t+1} = FFN(U_{w \leftarrow s}^{t+1} + H_w^t) \\ U_{s \leftarrow w}^{t+1} = GAT(H_s^t, H_w^{t+1}, H_w^{t+1}) \\ H_s^{t+1} = FFN(U_{s \leftarrow w}^{t+1} + H_s^t) \end{aligned} \tag{9}$$

Model	PubMed			arXiv		
	R-1	R-2	R-L	R-1	R-2	R-L
Oracle (Xiao and Carenini, 2020)	55.05	27.48	49.11	53.89	23.07	46.54
SummaRuNNer^{+}	43.89	18.78	30.36	42.91	16.65	28.53
Seq2seq-attentive^{+}	44.81	19.74	31.48	43.58	17.37	29.30
Cheng&Lapata (2016)$^{+}$	43.89	18.53	30.17	42.24	15.97	27.88
Discourse-aware*	38.93	15.37	35.21	35.80	11.05	31.80
ExtSum-LG (Xiao and Carenini, 2020)	45.39	20.37	40.99	44.01	17.79	39.09
Match-Sum (Zhong et al., 2020)	41.21	19.41	36.75	40.59	12.98	32.64
Topic-GraphSum (Cui and Hu, 2021)	45.95	20.81	33.97	44.03	18.52	32.41
SSN-DM (Cui and Hu, 2021)	46.73	21.00	34.10	45.03	**19.03**	32.58
MTGNN-SUM	**48.42**	**22.26**	**43.66**	**46.39**	18.58	**40.50**

Table 1: Results on PubMed and arXiv datasets. 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. Oracle indicates the ground truth results by using the greedy algorithm, which is regarded as the upper bound. Our results are reported by averaging values of 3 runs.

2.3 Multi Graph Neural Network for EDS

The outputs of sentence features from the two aforementioned layers are then concatenated for the final representation, which is formulated as follows:

$$H = H_s^{G_1} \oplus H_s^{G_2} \tag{10}$$

Observably, by concatenating the outputs of the two aforementioned graph layers, the final representation includes the information of both inter and intra-sentence relations. Sequentially, the output of the concatenation is put into a sentence classifier for ranking the classification.

3 Experiments

3.1 Datasets

Two long document datasets are taken into account for the evaluation. Specifically, PubMed and arXiv are standard datasets for long documents, which are scientific papers. For the data processing, we use the same split as the work in Cohan et al. (2018) to process the arXiv and PubMed datasets for the evaluation and follow Liu and Lapata (2019) to get ground-truth labels.

3.2 Hyperparameter Setting

Regarding the encoding, the vocabulary is limited to 50,000 and the tokens are initialized with 300-dimensional with Glove embedding. The dimension of sentence node and edge features are set to 128 and 50, respectively. The number of multi-head in each GAT layer is set to 8. For the document encoder, we use the *bert-base-uncased* version of BERT and fine-tune for the experiments. In the case of the decoding process, we select top-6 and top-5 for PubMed and arXiv datasets, respectively, according to the best performance of the validation set. Due to the limited computational resources, the maximum number of sentences in each document is set to 150, which means only the first 150 sentences in each document are taken into account. More analysis of the length of sentences is presented in the following section. The model is trained with the Adam optimizer. The learning rate is set to 1e-3 and use early stop with every three epochs. Moreover, learning rate decay is used after each epoch to improve performance. All models are trained on a single Tesla V100 32GB GPU, which has completed the training process with around 10 epochs. The total time for each epoch with the best model is around 6 hours and 3 hours for PubMed and arXiv datasets, respectively.

3.3 Experimental Results

Table 1 shows the results of our method compared with state-of-the-art models on PubMed and arXiv, respectively. The comparison models are divided into different parts. The first part reports the results of Oracle, which is regarded as ground truth extracted sentences. The second part shows the results of the approach without pre-trained language models. The third approach includes BERT-based models. The next section presents the result of the graph-based approach including the models with the document-level approach, which requires different levels of information such as words, sentences,

latent topics, and spotlights redundancy dependencies between sentences. The last section is our model, which is named MTGNN-SUM.

Accordingly, our results outperform state-of-the-art models in this research field. In particular, only R-2 of SSN-DM, the lasted state-of-the-art model is slightly better than our method in the case of arXiv datasets. However, the R-L metric of our method is much higher than the SSN-DM model. The results indicate the advantage of the proposed method by integrating both inter and intra-sentence relations. Specifically, inter-sentence allows information to flow for all sentence nodes and intra-sentence enriches the information of sentence nodes that contain common words. This issue especially is able to deal with the long-range dependency problem because the sentences, which are far from each other (e.g., by the distance of sentence positions), are able to share the information by using common words. Notably, our model does not need to consider any external semantic nodes for enriching global information (e.g., latent topic).

4 Quality Analysis

Ablation Study. In our model, we enrich the complex relationships by exploring both heterogeneous graph and homogeneous graph operations for sentence connection. In order to explore the effectiveness of each component, we design different variants of the proposed model as follows:

- **HomoGraph-SUM**: contains a graph attention layer for document encoding to extract inter-sentence relationships. The model is constructed following the description in Section 2.1 of Homogeneous Graph Neural Network.
- **HeterGraph-SUM**: contains a heterogeneous graph layer that contains semantic nodes to enrich the cross-sentence relations. Specifically, HeterGraph-Sum is designed following the description in Section 2.2.

The results of those aforementioned variants of our model on benchmark datasets are presented in Tab. 2. Accordingly, MTGNN-SUM outperforms all variants, which proves that executing message passing across sentences in the proposed model by incorporating both graph structures can achieve better results.

Length of Document. In this study, we set the maximum number of sentences in each document to equal 150 due to our limited computational resources. Though, we are able to improve the performance by learning whole-length sentences of the datasets, which include many documents with more than 200 sentences. In order to evaluate the importance of the document length value, we tested our model with the maximum number of sentences being 50 and 100 sentences, respectively. The results of the test models on different values of maximum document sizes are shown in Table 3. Accordingly, by increasing the maximum length of sentences, the performances are improved. In particular, the results indicated that tuning the max length of sentence value is able to enhance the performance. Specifically, we take this issue into account for the future work of this study by executing our model with a longer maximum size of documents.

Dataset	Model	R-1	R-2	R-L
PubMed	HomoGraph-SUM	39.29	13.74	34.49
	HeterGraph-SUM	46.03	19.79	41.48
	MTGNN-SUM	48.42	22.26	43.66
arXiv	HomoGraph-SUM	41.13	13.11	35.84
	HeterGraph-SUM	45.06	16.97	39.38
	MTGNN-SUM	46.39	18.58	40.50

Table 2: Ablation study results on two datasets.

Dataset	Model	R-1	R-2	R-L
PubMed	MTGNN-SUM-50	46.20	20.04	41.58
	MTGNN-SUM-100	47.85	21.64	43.13
	MTGNN-SUM-150	48.42	22.26	43.66
arXiv	MTGNN-SUM-50	44.91	16.89	39.14
	MTGNN-SUM-100	46.09	17.98	40.29
	MTGNN-SUM-150	46.39	18.58	40.50

Table 3: Results of the proposed model with different lengths of sentences.

5 Conclusion

This paper presents a novel graph-based method for the EDS task, which focuses on exploiting the complex relationship in terms of both inter and intra-sentence relations of the long-form documents. Specifically, two types of graph structures are developed for enriching sentence representations. The experiments on two benchmark datasets show promising results of the proposed method. Regarding future work, segmentation methods are taken into account for dividing long documents into paragraphs. Specifically, analyzing the complex relationships between paragraphs and integrating them into graphs as an additional node is able to enrich the information for the representation.

References

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

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, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. 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.

Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631, Online. 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)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. 2021. Graph neural networks for natural language processing: A survey. *CoRR*, abs/2106.06090.

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)*, pages 3011–3021, Hong Kong, China. 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*, pages 516–528, Suzhou, China. 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*.

Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 793–803. ACM.

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.

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*, pages 6197–6208, Online. Association for Computational Linguistics.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.