HETFORMER: Heterogeneous Transformer with Sparse Attention for Long-Text Extractive Summarization

Ye Liu[1], **Jian-Guo Zhang**[1], **Yao Wan**[2], **Congying Xia**[1], **Lifang He**[3], **Philip S. Yu**[1]
[1] University of Illinois at Chicago, Chicago, IL, USA
[2] Huazhong University of Science and Technology, Huhan, China
[3] Lehigh University, Bethlehem, PA, USA
{yliu279, jzhan51, cxia8, psyu@}@uic.edu,
wanyao@hust.edu.cn, lih319@lehigh.edu

Abstract

To capture the semantic graph structure from raw text, most existing summarization approaches are built on GNNs with a pre-trained model. However, these methods suffer from cumbersome procedures and inefficient computations for long-text documents. To mitigate these issues, this paper proposes HETFORMER, a Transformer-based pre-trained model with multi-granularity sparse attentions for long-text extractive summarization. Specifically, we model different types of semantic nodes in raw text as a potential heterogeneous graph and directly learn heterogeneous relationships (edges) among nodes by Transformer. Extensive experiments on both single- and multi-document summarization tasks show that HETFORMER achieves state-of-the-art performance in Rouge F1 while using less memory and fewer parameters.

1 Introduction

Recent years have seen a resounding success in the use of graph neural networks (GNNs) on document summarization tasks (Wang et al., 2020; Hanqi Jin, 2020), due to their ability to capture inter-sentence relationships in complex document. Since GNN requires node features and graph structure as input, various methods, including extraction and abstraction (Li et al., 2020; Huang et al., 2020; Jia et al., 2020), have been proposed for learning desirable node representations from raw text. Particularly, they have shown that Transformer-based pre-trained models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) offer an effective way to initialize and fine tune the node representations as the input of GNN.

Despite great success in combining Transformer-based pre-trained models with GNNs, all existing approaches have their limitations. The first limitation lies in the adaptation capability to long-text input. Most pre-trained methods truncate longer documents into a small fixed-length sequence (e.g., $n = 512$ tokens), as its attention mechanism requires a quadratic cost w.r.t. sequence length. This would lead to serious information loss (Li et al., 2020; Huang et al., 2020). The second limitation is that they use pre-trained models as a multi-layer feature extractor to learn better node features and build multi-layer GNNs on top of extracted features, which have cumbersome networks and tremendous parameters (Jia et al., 2020).

Recently there have been several works focusing on reducing the computational overhead of fully-connected attention in Transformers. Especially, ETC (Ravula et al., 2020) and Longformer (Beltagy et al., 2020) proposed to use local-global sparse attention in pre-trained models to limit each token to attend to a subset of the other tokens (Child et al., 2019), which achieves a linear computational cost of the sequence length. Although these methods have considered using local and global attentions to preserve hierarchical structure information contained in raw text data, their abilities are still not enough to capture multi-level granularities of semantics in complex text summarization scenarios.

In this work, we propose HETFORMER, a HETerogeneous transFORMER-based pre-trained model for long-text extractive summarization using multi-granularity sparse attentions. Specifically, we treat tokens, entities, sentences as different types of nodes and the multiple sparse masks as different types of edges to represent the relations (e.g., token-to-token, token-to-sentence), which can preserve the graph structure of the document even with the raw textual input. Moreover, our approach will eschew GNN and instead rely entirely on a sparse attention mechanism to draw heterogeneous graph structural dependencies between input tokens.

The main contributions of the paper are summarized as follows: 1) we propose a new structured pre-trained method to capture the heterogeneous structure of documents using sparse attention; 2) we extend the pre-trained method to longer text

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 146–154
November 7–11, 2021.

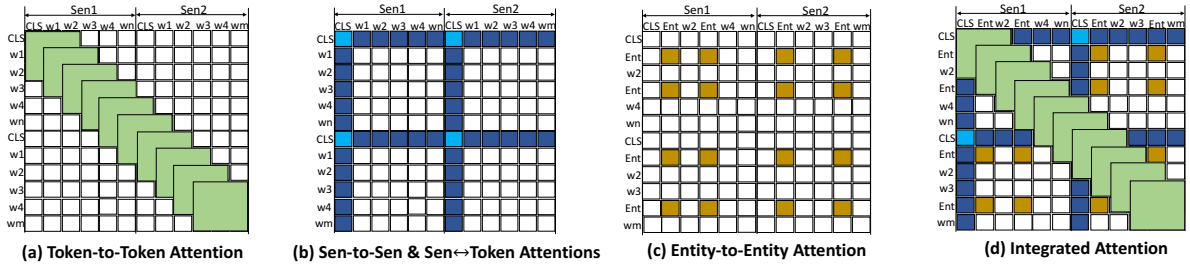

Figure 1: An illustration of sparse attention patterns ((a), (b), (c)) and their combination (d) in HETFORMER.

extractive summarization instead of truncating the document to small inputs; 3) we empirically demonstrate that our approach achieves state-of-the-art performance on both single- and multi-document extractive summarization tasks.

2 HETFORMER on Summarization

HETFORMER aims to learn a heterogeneous Transformer in pre-trained model for text summarization. To be specific, we model different types of semantic nodes in raw text as a potential heterogeneous graph, and explore multi-granularity sparse attention patterns in Transformer to directly capture heterogeneous relationships among nodes. The node representations will be interactively updated in a fine-tuned manner, and finally, the sentence node representations are used to predict the labels for extractive text summarization.

2.1 Node Construction

In order to accommodate multiple granularities of semantics, we consider three types of nodes: *token*, *sentence* and *entity*.

The *token* node represents the original textual item that is used to store token-level information. Different from HSG (Wang et al., 2020) which aggregates identical tokens into one node, we keep each token occurrence as a different node to avoid ambiguity and confusion in different contexts. Each *sentence* node corresponds to one sentence and represents the global information of one sentence. Specifically, we insert an external `[CLS]` token at the start of each sentence and use it to encode features of each tokens in the sentence. We also use the interval segment embeddings to distinguish multiple sentences within a document, and the position embeddings to display monotonical increase of the token position in the same sentence. The *entity* node represents the named entity associated with the topic. The same entity may appear in multiple spans in the document. We utilize NeuralCoref[1] to obtain the coreference resolution of each entity, which can be used to determine whether two expressions (or "mentions") refer to the same entity.

2.2 Sparse Attention Patterns

Our goal is to model different types of relationships (edges) among nodes, so as to achieve a sparse graph-like structure directly. To this end, we leverage multi-granularity sparse attention mechanisms in Transformer, by considering five attention patterns, as shown in Fig. 1: *token-to-token* (*t2t*), *token-to-sentence* (*t2s*), *sentence-to-token* (*s2t*), *sentence-to-sentence* (*s2s*) and *entity-to-entity* (*e2e*).

Specifically, we use a fixed-size window attention surrounding each token (Fig. 1(a)) to capture the short-term *t2t* dependence of the context. Even if each window captures the short-term dependence, by using multiple stacked layers of such windowed attention, it could result in a large receptive field (Beltagy et al., 2020). Because the top layers have access to all input locations and have the capacity to build representations that incorporate information across the entire input.

The *t2s* represents the attention of all tokens connecting to the sentence nodes, and conversely, *s2t* is the attention of sentence nodes connecting to all tokens across the sentence (the dark blue lines in Fig. 1(b)). The *s2s* is the attention between multiple sentence nodes (the light blue squares in Fig. 1(b)). To compensate for the limitation of *t2t* caused by using fixed-size window, we allow the *sentence* nodes to have unrestricted attentions for all these three types. Thus tokens that are arbitrarily far apart in the long-text input can transfer information to each other through the *sentence* nodes.

Complex topics related to the same entity may span multiple sentences, making it challenging for existing sequential models to fully capture the se-

[1] https://github.com/huggingface/neuralcoref

mantics among entities. To solve this problem, we introduce the *e2e* attention pattern (Fig. 1(c). The intuition is that if there are several mentions of a particular entity, all the pairs of the same mentions are connected. In this way, we can facilitate the connections of relevant entities and preserve global context, e.g., entity interactions and topic flows.

Linear Projections for Sparse Attention. In order to ensure the sparsity of attention, we create three binary masks for each attention patterns $\mathbf{M}^{t2t}$, $\mathbf{M}^{ts}$ and $\mathbf{M}^{e2e}$, where 0 means disconnection and 1 means connection between pairs of nodes. In particular, $\mathbf{M}^{ts}$ is used jointly for *s2s*, *t2s* and *s2t*. We use different projection parameters for each attention pattern in order to model the heterogeneity of relationships across nodes. To do so, we first calculate each attention with its respective mask and then sum up these three attentions together as the final integrated attention (Fig. 1(d)).

Each sparse attention is calculated as: $\mathbf{A}^m = \text{softmax}\left(\frac{\mathbf{Q}^m \mathbf{K}^{m\top}}{\sqrt{d_k}}\right)\mathbf{V}^m, m \in \{t2t, ts, e2e\}$. The query $\mathbf{Q}^m$ is calculated as $(\mathbf{M}^m \odot \mathbf{X})\mathbf{W}_Q^m$, where $\mathbf{X}$ is the input text embedding, $\odot$ represents the element-wise product and $\mathbf{W}_Q^m$ is the projection parameter. The key $\mathbf{K}^m$ and the value $\mathbf{V}^m$ are calculated in a similar way as $\mathbf{Q}^m$, but with respect to different projection parameters, which are helpful to learn better representation for heterogeneous semantics. The expensive operation of full-connected attention is $\mathbf{Q}\mathbf{K}^T$ as its computational complexity is related to the sequence length (Kitaev et al., 2020). While in HETFORMER, we follow the implementation of Longformer that only calculates and stores attention at the position where the mask value is 1 and this results in a linear increase in memory use compared to quadratic increase for full-connected attention.

2.3 Sentence Extraction

As extractive summarization is more general and widely used, we build a classifier on each *sentence* node representation o_s to select sentences from the last layer of HETFORMER. The classifier uses a linear projection layer with the activation function to get the prediction score for each sentence: $\tilde{y}_s = \sigma\left(o_s \mathbf{W}_o + b_o\right)$, where σ is the sigmoid function, $\mathbf{W}_o$ and b_o are parameters of projection layer.

In the training stage, these prediction scores are trained learned on the binary cross-entropy loss with the golden labels y. In the inference stage, these scores are used to sort the sentences and select the top-k as the extracted summary.

2.4 Extension to Multi-Document

Our framework can establish the document-level relationship in the same way as the sentence-level, by just adding *document* nodes for multiple documents (i.e., adding the `[CLS]` token in front of each document) and calculate the *document*↔*sentence* (*d2s*, *s2d*), *document*↔*token* (*d2t*, *t2d*) and *document-to-document* (*d2d*) attention patterns. Therefore, it can be easily adapted from the single-document to multi-document summarization.

2.5 Discussions

The most relevant approaches to this work are Longformer (Beltagy et al., 2020) and ETC (Ravula et al., 2020) which use a hierarchical attention pattern to scale Transformers to long documents. Compared to these two methods, we formulate the Transformer as multi-granularity graph attention patterns, which can better encode heterogeneous node types and different edge connections. More specifically, Longformer treats the input sequence as one sentence with the single tokens marked as global. In contrast, we consider the input sequence as multi-sentence units by using *sentence-to-sentence* attention, which is able to capture the inter-sentence relationships in the complex document. Additionally, we introduce *entity-to-entity* attention pattern to facilitate the connection of relevant subjects and preserve global context, which are ignored in both Longformer and ETC. Moreover, our model is more flexible to be extended to the multi-document setting.

3 Experiments

3.1 Datasets

CNN/DailyMail is the most widely used benchmark dataset for single-document summarization (Zhang et al., 2019; Jia et al., 2020). The standard dataset split contains 287,227/13,368/11,490 samples for train/validation/test. To be comparable with other baselines, we follow the data processing in (Liu and Lapata, 2019b; See et al., 2017).

Multi-News is a large-scale dataset for multi-document summarization introduced in (Fabbri et al., 2019), where each sample is composed of 2-10 documents and a corresponding human-written summary. Following Fabbri et al. (2019), we split the dataset into 44,972/5,622/5,622 for train/validation/test. The average length of source

documents and output summaries are 2,103.5 tokens and 263.7 tokens, respectively. Given the N input documents, we taking the first L/N tokens from each source document. Then we concatenate the truncated source documents into a sequence by the original order. Due to the memory limitation, we truncate input length L to 1,024 tokens. But if the memory capacity allows, our model can process the max input length = 4,096.

While the dataset contains abstractive gold summaries, it is not readily suited to training extractive models. So we follow the work of (Zhou et al., 2018) on extractive summary labeling, constructing gold-label sequences by greedily optimizing R-2 F1 on the gold-standard summary.

3.2 Baselines and Metrics

We evaluate our proposed model with the pre-trained language model (Devlin et al., 2018; Liu et al., 2019), the state-of-the-art GNN-based pre-trained language models (Wang et al., 2020; Jia et al., 2020; Hanqi Jin, 2020) and pre-trained language model with the sparse attention (Narayan et al., 2020; Beltagy et al., 2020). And please check Appendix B for the detail.

We use unigram, bigram, and longest common subsequence of Rouge F1 (denoted as R-1, R-1 and R-L) (Lin and Och, 2004)[2] to evaluate the summarization qualities. Note that the experimental results of baselines are from the original papers.

3.3 Implementation Detail

Our model HETFORMER[3] is initialized using the Longformer pretrained checkpoints `longformer-base-4096`[4], which is further pertained using the standard masked language model task on the Roberta checkpoints `roberta-base`[5] with the documents of max length 4,096. We apply dropout with probability 0.1 before all linear layers in our models. The proposed model follows the Longformer-base architecture, where the number of d_{model} hidden units in our models is set as 768, the d_h hidden size is 64, the layer number is 12 and the number of heads is 12. We train our model for 500K steps on the TitanRTX, 24G GPU with gradient accumulation in every two steps with Adam optimizers.

[2] https://pypi.org/project/rouge/
[3] https://github.com/yeliu918/HETFORMER
[4] https://github.com/allenai/longformer
[5] https://github.com/huggingface/transformers

Model	R-1	R-2	R-L
HiBERT (Zhang et al., 2019)	42.31	19.87	38.78
HSG (Wang et al., 2020)	42.95	19.76	39.23
HAHsum$_{\text{Large}}$ (Jia et al., 2020) *	**44.67**	**21.30**	**40.75**
MatchSum (Zhong et al., 2020)	44.41	20.86	40.55
BERT$_{\text{Base}}$ (Devlin et al., 2018)	41.55	19.34	37.80
RoBERTa$_{\text{Base}}$ (Liu et al., 2019)	42.99	20.60	39.21
ETC$_{\text{Base}}$ (Narayan et al., 2020)	43.43	20.54	39.58
Longformer$_{\text{Base}}$ (Beltagy et al., 2020)	43.20	20.38	39.61
HETFORMER$_{\text{Base}}$	44.55	20.82	40.37

Table 1: Rouge F1 scores on test set of CNN/DailyMail. *Note that HAHsum$_{\text{Large}}$ uses large verision while the proposed model is based on the base version.

Model	R-1	R-2	R-L
HiBERT (Zhang et al., 2019)	44.32	15.11	29.26
Hi-MAP (Fabbri et al., 2019)	45.21	16.29	41.39
HDSG (Wang et al., 2020)	46.05	16.35	42.08
MatchSum (Zhong et al., 2020)	46.20	16.51	41.89
MGsum$_{\text{Base}}$ (Hanqi Jin, 2020)	45.04	15.98	-
Graphsum$_{\text{Base}}$ (Li et al., 2020)	46.07	17.42	-
Longformer$_{\text{Base}}$ (Beltagy et al., 2020)	45.34	16.00	40.54
HETFORMER$_{\text{Base}}$	**46.21**	**17.49**	**42.43**

Table 2: Rouge F1 scores on test set of Multi-News. '-' means that the original paper did not report the result.

Learning rate schedule follows the strategies with warming-up on first 10,000 steps (Vaswani et al., 2017). We select the top-3 checkpoints according to the evaluation loss on validation set and report the averaged results on the test set.

For the testing stage, we select top-3 sentences for CNN/DailyMail and top-9 for Multi-News according to the average length of their human-written summaries. Trigram blocking is used to reduce repetitions.

3.4 Summerization Results

As shown in Table 1, our approach outperforms or is on par with current state-of-the-art baselines. Longformer and ETC outperforms the hierarchical structure model using fully-connected attention model HiBERT, which shows the supreme of using sparse attention by capturing more relations (e.g., token-to-sentence and sentence-to-token). Comparing to the pre-trained models using sparse attention, HETFORMER considering the heterogeneous graph structure among the text input outperforms Longformer and ETC. Moreover, HETFORMER achieves competitive performance compared with GNN-based models, such as HSG and HAHsum. Our model is slightly lower than the performance of HAHsum$_{\text{large}}$. But it uses large architecture (24 layers with about 400M parameters), while our

	BERT	RoBERTa	Longformer	Ours
Memory Cost	3,057M	3,540M	1,650M	1,979M

Table 3: Memory cost of different pre-trained models

model builds on the base model (12 layers with about 170M parameters). Table 2 shows the results of multi-document summarization. Our model outperforms all the extractive and abstractive baselines. These results reveal the importance of modeling the longer document to avoid serious information loss.

3.5 Memory Cost

Compared with the self-attention component requiring quadratic memory complexity in original Transformers, the proposed model only calculates the position where attention pattern mask=1, which can significantly save the memory cost. To verify that, we show the memory costs of BERT, RoBERTa, Longformer and HETFORMER base-version model on the CNN/DailyMail dataset with the same configuration (input length = 512, batch size = 1).

From the results in Table 3, we can see that HETFORMER only takes 55.9% memory cost of RoBERTa model and also does not take too much more memory than Longformer.

3.6 Ablation Study

To show the importance of the design choices of our attention patterns, we tried different variants and reported their controlled experiment results. To make the ablation study more manageable, we train each configuration for 500K steps on the single-document CNN/DailyMail dataset, then report the Rouge score on the test set.

The top of Table 4 demonstrates the impact of different ways of configuring the window sizes per layer. We observe that increasing the window size from the bottom to the top layer leads to the best performance (from 32 to 512). But the reverse way leads to worse performance (from 512 to 32). And using a fixed window size (the average of window sizes of the other configuration) leads to a performance that it is in between.

The middle of Table 4 presents the impact of incorporating the sentence node in the attention pattern. In implementation, no sentence node means that we delete the `[CLS]` tokens of the document input and use the average representation of each token in the sentences as the sentence representation. We observe that without using the sentence node to fully connect with the other tokens could decrease the performance.

The bottom of Table 4 shows the influence of using the entity node. We can see that without the entity node, the performance will decrease. It demonstrates that facilitating the connection of relevant subjects can preserve the global context, which can benefit the summarization task.

Model	R-1	R-2	R-L
Decreasing w (from 512 to 32)	43.98	20.33	39.39
Fixed w (=128)	43.92	20.43	39.43
Increasing w (from 32 to 512)	**44.55**	**20.82**	**40.37**
No Sentence node	42.15	20.12	38.91
No Entity node	43.65	20.40	39.28

Table 4: Top: changing window size across layers. Middle: entity-to-entity attention pattern influence. Bottom: sentence-to-sentence attention pattern influence

4 Conclusion

For the task of long-text extractive summarization, this paper has proposed HETFORMER, using multi-granularity sparse attention to represent the heterogeneous graph among texts. Experiments show that the proposed model can achieve comparable performance on a single-document summarization task, as well as state-of-the-art performance on the multi-document summarization task with longer input document. In our future work, we plan to expand the edge from the binary type (connect or disconnect) to more plentiful semantic types, i.e., *is-a*, *part-of*, and others (Zhang et al., 2020).

5 Acknowledgements

We would like to thank all the reviewers for their helpful comments. This work is supported by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

References

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

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep

bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of ssociation for Computational Linguistics*, pages 4171–4186.

Alexander R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the Conference of Association for Computational Linguistics*, pages 1074–1084.

Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. 2021. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of ACM International Conference on Information and Knowledge Management*.

Xiaojun Wan Hanqi Jin, Tianming Wang. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the Conference of Association for Computational Linguistics*, pages 6244–6254.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the Web Conference*, pages 2704–2710.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *Proceedings of the Conference of Association for Computational Linguistics*, page 5094–5107.

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 Conference of Neural Information Processing Systems*, pages 3622–3631.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *Proceedings of the International Conference on Learning Representations*.

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 Conference of Association for Computational Linguistics*, pages 6232—6243.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the Conference of Association for Computational Linguistics*, pages 605–612.

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *Proceedings of the Conference of Association for Computational Linguistics*, pages 5070–5081.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *Proceedings of the Conference of Neural Information Processing Systems*, pages 3730–3740.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S Yu. 2020. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ye Liu, Yao Wan, Jian-Guo Zhang, Wenting Zhao, and Philip S Yu. 2021. Enriching non-autoregressive transformer with syntactic and semanticstructures for neural machine translation. In *Proceedings of the European Chapter of the Association for Computational Linguistics*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shashi Narayan, Joshua Maynez, Jakub Adamek, Daniele Pighin, Blaž Bratanič, and Ryan McDonald. 2020. Stepwise extractive summarization and planning with structured transformers. In *Proceedings of the Conference of Neural Information Processing Systems*, page 4143–4159.

Anirudh Ravula, Chris Alberti, Joshua Ainslie, Li Yang, Philip Minh Pham, Qifan Wang, Santiago Ontanon, Sumit Kumar Sanghai, VConference of Association for Computational Linguisticsav Cvicek, and Zach Fisher. 2020. Etc: Encoding long and structured inputs in transformers. In *Proceedings of the Conference of Neural Information Processing Systems*, pages 268–284.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the Conference of Association for Computational Linguistics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Conference of Neural Information Processing Systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. In *Proceedings of the International Conference on Learning Representations*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous

graph neural networks for extractive document summarization. In *Proceedings of the Conference of Association for Computational Linguistics*, page 6209–6219.

Shaowei Yao, Tianming Wang, and Xiaojun Wan. 2020. Heterogeneous graph transformer for graph-to-sequence learning. In *Proceedings of the Conference of Association for Computational Linguistics*, pages 7145–7154.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. In *Proceedings of the Conference of Neural Information Processing Systems*, pages 11983–11993.

Li Zhang, Yan Ge, and Haiping Lu. 2020. Hop-hop relation-aware graph neural networks. *arXiv preprint arXiv:2012.11147*.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the Conference of Association for Computational Linguistics*, page 5059–5069.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the Conference of 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 Conference of Association for Computational Linguistics*, page 654–663.

A Background

A.1 Graph-enhanced Summarization

In the recent state-of-the-art summarization models, there is a trend to extract the structure from the text to formulate the document text as a hierarchical structure or heterogeneous graph (Liu et al., 2020). HiBERT (Zhang et al., 2019), GraphSum (Li et al., 2020) and HT (Liu and Lapata, 2019a) consider the word-level, sentence-level and document-level of the input text to formulate the hierarchical structure. MGSum (Hanqi Jin, 2020), ASGARD (Huang et al., 2020), HSG (Wang et al., 2020) and HAHSum (Jia et al., 2020) construct the source article as a heterogeneous graph where words, sentences, and entities are used as the semantic nodes and they iteratively update the sentence nodes representation which is used to do the sentence extraction.

The limitation of those models is that they use pre-trained methods as the feature-based model to learn the node feature and build GNN layers upon the node which brings more training parameters than just using pre-trained methods. Compared with those models, our work can achieve the same thing but using the lite framework. Moreover, these models typically limit inputs to $n = 512$ tokens since the $O(n^2)$ cost of attention. Due to the long source article, when applying BERT or RoBERTa to the summarization task, they need to truncate source documents into one or several smaller block input (Li et al., 2020; Jia et al., 2020; Huang et al., 2020).

A.2 Structure Transformer

Huang et al. (2021) proposed an efficient encoder-decoder attention with head-wise positional strides, which yields ten times faster than existing full attention models and can be scale to long documents. Liu et al. (2021) leveraged the syntactic and semantic structures of text to improve the Transformer and achieved nine times speedup. Our model focuses on the different direction to use graph-structured sparse attention to capture the long term dependence on the long text input. The most related approaches to the work presented in this paper are Longformer (Beltagy et al., 2020) and ETC (Ravula et al., 2020) which feature a very similar global-local attention mechanism and take advantage of the pre-trained model RoBERTa. Except Longformer has a single input sequence with some tokens marked as global (the only ones that use full attention), while the global tokens in the ETC is pre-trained with CPC loss. Comparing with those two works, we formulate the heterogeneous attention mechanism, which can consider the word-to-word, word-to-sen, sen-to-word and entity-to-entity attention.

A.3 Graph Transformer

With the great similarity between the attention mechanism used in both Transformer (Vaswani et al., 2017) and Graph Attention network (Veličković et al., 2017), there are several recent Graph Transformer works recently. Such as GTN (Yun et al., 2019), HGT (Hu et al., 2020), (Fan et al., 2021) and HetGT (Yao et al., 2020) formulate the different type of the attention mechanisms to capture the node relationship in the graph.

The major difference between of our work and Graph Transformer is that the input of graph transformer is structural input, such as graph or dependence tree, but the input of our HeterFormer is unstructured text information. Our work is to convert the transformer to structural structure so that it can capture the latent relation in the unstructured text, such as the word-to-word, word-to-sent, sent-to-word, sent-to-sent and entity-to-entity relations.

B Baseline Details

Extractive Models:

BERT (or RoBERTa) (Devlin et al., 2018; Liu et al., 2019) is a Transformer-based model for text understanding through masking language models. **HIBERT** (Zhang et al., 2019) proposed a hierarchical Transformer model where it first encodes each sentence using the sentence Transformer encoder, and then encoded the whole document using the document Transformer encoder. **HSG**, HDSG (Wang et al., 2020) formulated the input text as the heterogeneous graph which contains different granularity semantic nodes, (like word, sentence, document nodes) and connected the nodes with the TF-IDF. HSG used CNN and BiLSTM to initialize the node representation and updated the node representation by iteratively passing messages by Graph Attention Network (GAT). In the end, the final sentence nodes representation is used to select the summary sentence. **HAHsum** (Jia et al., 2020) constructed the input text as the heterogeneous graph containing the word, named entity, and sentence node. HAHsum used a pre-trained ALBERT to learn the node initial representation and then adapted GAT to iteratively learn node hidden repre-

sentations. **MGsum** (Hanqi Jin, 2020) treated documents, sentences, and words as the different granularity of semantic units, and connected these semantic units within a multi-granularity hierarchical graph. They also proposed a model based on GAT to update the node representation. **ETC** (Narayan et al., 2020), and Longformer (Beltagy et al., 2020) are two pre-trained models to capture hierarchical structures among input documents through the sparse attention mechanism.

Abstractive Models: Hi-MAP (Fabbri et al., 2019) expands the pointer-generator network model into a hierarchical network and integrates an MMR module to calculate sentence-level scores. **Graphsum** (Li et al., 2020) leverage the graph representations of documents by processing input documents as the hierarchical structure with a pre-trained language model to generate the abstractive summary.