GNN for Long Scientific Document Summarization

Xiangru Tang, Yizhi Zhao, Wenxin Xu, Chengqian Jin
{xiangru.tang, yizhi.zhao, wenxin.xu, chengqian.jin}@yale.edu

Abstract

By enriching the complex information between words and sentences, graph neural networks have recently been introduced as an emerging method for many natural language processing tasks. However, for most neural-based models, long scientific documents (e.g., scientific papers and medical papers) are truncated, which leads to the challenge of information loss of inter-sentence relationships. In this paper, we try to improve the performance of extractive document summarization of long scientific documents based on GNN. To address this issue, we propose a new approach that leverages the power of GNNs and pretrained language models. Specifically, BERT is considered for improving sentence information in layers. Experiments on two benchmark datasets with long documents such as PubMed and ArXiv show that our method outperforms state-of-the-art models in this research area.

1 Introduction

The purpose of text summarization is to automatically compress long documents into shorter versions, preserving a concise description of the content. Most of the previous work has focused on the news domain and achieved promising results using neural encoder-decoder architectures. Although text summarization systems have not been much explored in other fields such as scientific papers, they still have broad application prospects. Generating good abstracts for scientific papers is a very challenging task, even for humans.

Though transformer models such as BERT (Devlin et al., 2018), and other variants have achieved state-of-the-art results on many challenging Natural Language Processing (NLP) tasks. Sequence to sequence (seq2seq) models have been successfully used for a variety of NLP tasks, including text summarization, machine translation, and question answering (Keneshloo et al., 2018). While successful, Transformer-based models have limits on the length of the input sequence (Zhong et al., 2020). The quadratic computational and memory complexities of large transformers have limited their scalability for long document summarization as the token length for a standard transformer is limited to 512 tokens (Pang et al., 2022). When the input is long, the learning degrades particularly for tasks that require a comprehensive understanding of the entire paragraph or document. One of the main learning challenges for seq2seq models is that the decoder needs to attend to token-level representations from the encoder to predict the next token, while at the same time it must learn from a large context. Our project is designed to build efficient models to better handle long sequences, thus we evaluate it on the long document summarization task, which tends to have long source sequences as the input.

GNN models have been shown to effectively capture semantic relationships by modeling graph-structured representations between sentences (Zhou et al., 2020). In this study, we conduct a study to improve performance on extractive problems from long documents, the core idea of which is to exploit the complex relations of sentence connections. Specifically, we use the advantages of GNN in extracting semantic information between words and sentences (Zhang et al., 2019a), and employ Graph Attention Network with BERT for sentence representation to extract the relationship between sentences. Here, our proposed composition model is able to capture the semantic information of inter-sentence and intra-sentence connections. We propose a new method for learning complex relations in sentence connections. We evaluate our proposed method using two benchmark long document datasets, PubMed and ArXiv. Furthermore, our proposed method is able to extract complex relations of intra-sentence and inter-sentence relations, which can be easily extended to other NLP tasks.

2 Related Work

Transformers are successful in many natural language processing areas, which include text summarization. Zhang, Wei, and Zhou (2019b) proposed HIBERT, Hierarchical Bidirectional Encoder Representations from Transformers. This model has a sentence encoder that transforms a sentence into a vector and a document encoder that learns sentence representations by surrounding context. Both of them are based on transformer and are nested in a hierarchical way to learn the vector representation of the document. The model is pretrained by sentence predicting and the summarization task is modeled to label each sentence as summary or not. This model achieved state-of-the-art results in two datasets, CNNDM and GIGA-CM. Nevertheless, this model has limitation on input document size as 512 words. Longer documents need to be split into blocks and feed into the model multiple times, which means loss of information between blocks. Besides, extractive models like this have shortages in nature for documents that do not contain perfect summary sentence, due to their lack of ability to rephrase.

LongT5, Efficient Text-to-Test Transformer for Long Sequences, which allows for scaling both input length and model size simultaneously (Guo et al., 2021). This model introduces an attention mechanism called Transient Global (TGlobal), which echos local/global attention mechanism of a family of long-input transformer - Extended Transformer Construction (ETC) (Ainslie et al., 2020) and serves as a drop-in replacement to vanilla attention in T5 architecture (Raffel et al., 2019), and hence additional side-inputs are not required. The keypoint of local/global attention mechanism is to lower the quadratic cost when scaling to long inputs by bringing in local sparsity. Furthermore, PEGASUS-style Principle Sentences Generation pre-training strategies were adopted, resulting in SOA performance on several datasets: arXiv, PubMed, BigPatent, MediaSum and TriviaQA.

Graph neural networks have also been previously applied onto text summarization tasks. One example is the heterogeneous graph neural network (Wang et al., 2020). Its architecture consists of two types of nodes, one is the semantic node representing words, one is the super node representing discourses (sentences, phrases, or documents). Super nodes connect to semantic nodes that they contains and the edges weigh by the importance of words in the discourses (e.g. TF-IDF). Such design constructs the relationship between discourses by semantic units. Its output is discourses labels that indicate whether they will be in the summarization. This model is the first to introduce different types of nodes in graph neural networks for text summarization. The framework consists of three major parts: graph initializers which encodes nodes and edges for the document graph, the heterogeneous graph layer which updates nodes representations via message passing, and the sentence selector which predicts labels from a sequence of sentence node representation. The limitation was that the complicated inter-sentence relationship was not considered, especially the redundancy within extracted sentences. (Huang and Kurohashi, 2021) further extends this heterogeneous model. They proposed the model with three types of nodes, sentence nodes, EDU nodes, which are derived by segmenting the document into sub-sentential EDUs, and entity nodes, which are derived by clustering mentions in document. Among them, EDU nodes lie on the core. The discourse dependency between EDU nodes are based on RST tree of the document, the dependency between EDU nodes and entity nodes are based on coreference relations, and the dependency between sentence nodes and EDU nodes are based on constituent relations. Different types of nodes means different granularity to extract information from the document. Therefore, heterogeneous graph neural networks may have better representing capability compared to homogeneous network.

3 Model

Specifically, we leverage the capabilities of the graph attention network and BERT to exploit the complex relations of sentence connections. The main components of our method are presented in turn in the following sections.

For the document encoder, we use a BERT-style model to initialize the representation of sentence nodes in the text graph. We get these representations to use as input to the graph attention network. Then, we use the representation of the sentence to iteratively update based on the graph structure using GNN, the output of which is regarded as the final representation of the node and sent to the softmax classifier for sentence selection. In this way, we are able to take advantage of the complementary strengths of pretrained and graphical models. BERT is used to generate local hidden represen-

Datasets	**Source**	**# Pairs**			**Doc. Length**		**Sum. Length**		**# Sections**
		Train	Val	Test	# Words	# Sent.	# Words	# Sent.	
CNN	News	90,266	1,220	1,093	760.5	34.0	45.7	3.6	-
DailyMail	News	196,961	12,148	10,397	653.3	29.3	54.7	3.9	-
ScisummNet	Scientific Papers	1009	–	–	4203.4	178.0	150.7	7.4	6.5
arXiv†	Scientific Papers	215,913	6440	6436	4938.0	206.3	220.0	9.6	5.9
PubMed†	Scientific Papers	119,924	6633	6658	3016.0	86.4	203.0	6.9	5.6

Table 1: Dataset statistics. The dataset with † is what we are using.

tations between sentences. Insert CLS and SEP tokens at the beginning and end of each sentence in turn. Then, feed all tokens into BERT to learn the hidden state, which can be represented as follows:

$$h_1, h_2, ..., h_n = -\mathrm{BERT}(x_1, x_2, .., x_n), \quad (1)$$

Sentence relationships can be represented by a graph $\mathcal{G}$ with n nodes, where each node represents a word in the sentence. The edges of $\mathcal{G}$ represent dependencies between words. The neighbor nodes of node i can be represented by $\mathcal{N}_i$. The graph attention network iteratively updates each node representation (e.g., word embeddings) by aggregating neighborhood node representations using multi-head attention:

$$h^{l+1}_{att_i} = ||^{K}_{k=1} \sum_{j \in \mathcal{N}_i} \alpha^{lk}_{ij} W^l_k h^l_j \quad (2)$$

$$\alpha^{lk}_{ij} = attention(i, j) \quad (3)$$

where $h^{l+1}_{att_i}$ is the attention head of node i at layer $l+1$, $||^K_{k=1} x_i$ denotes the concatenation of vectors from x_1 to x_k, α^{lk}_{ij} is a normalized attention coefficient computed by the k-th attention at layer l, W^l_k is an input transformation matrix. In this paper, we adopt dot-product attention for $attention(i, j)$. Here, dot product has fewer parameters but similar performance with feedforward neural network.

At the end, we perform a node classification method on the sentences, sorting by score to select the sentences for the final summary. We use cross-entropy loss to classify sentences.

4 Results

We will utilize the existing dataset, the PubMed and arXiv datasets, which contain scientific articles from PubMed and arXiv respectively, and will use the abstract of the articles as the target summary. For the evaluation of our experiments, we will use classic evaluation metrics of ROUGE-1, ROUGE-2, and ROUGE-L. File2rouge will be used to evaluate Rouge score for all the results.

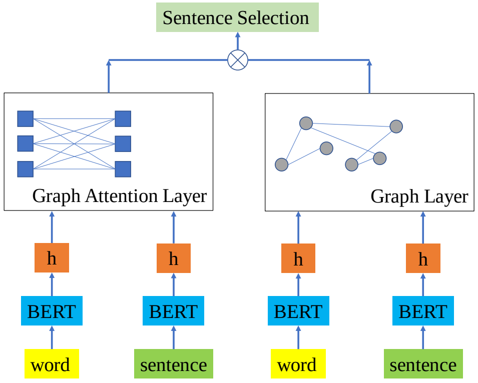

Figure 1: An illustration of our proposed model.

Our models were trained on a 4 Nvidia 3090 GPUs with a batch size of 8. We train all of our models using Adagrad with 0.15 learning rate and have an accumulator of 0.1. During training we are regularly measuring the loss and the ROUGE-1 F-score on the validation set of the dataset in order to monitor the learning of our model. We end the training when the validation loss stops improving. The results are listed here:

The overall performance of all models is shown in the table, from which several observations can be noted. First, our model outperforms most baseline models. Furthermore, the BERT SUM model and Transformer model can significantly outperform all existing baselines, demonstrating the power of pretrained models on this task. Nonetheless, after incorporating our GNN parts, this powerful model was further improved and reached a new state of the art. These results demonstrate the effectiveness of our model in capturing sentence relations for extractive summarization.

5 Conclusions

This paper proposes a novel graph-based approach to address the challenge of extractive summariza-

Dataset	arXiv			PubMed		
Systems	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
LEAD	28.29	5.99	24.84	33.89	9.93	29.70
Extractive						
TextRank	36.36	9.67	32.72	39.19	13.89	34.59
TransformerExt	43.14	13.68	38.65	42.39	18.37	38.99
BertSumExt	42.41	13.10	37.97	41.21	19.41	36.75
Abstractive						
PTGen	32.84	9.28	27.59	35.86	10.22	29.69
TransformerAbs	37.78	9.59	34.21	43.89	18.53	30.17
BertSumAbs	41.22	13.31	37.22	42.01	16.79	27.09
Our Model						
GNN-GAT-BERT	46.73	21.00	34.10	45.03	19.03	32.58

tion of long scientific documents. Our proposed method focuses on capturing the complex relations of inter-sentence and intra-sentence connections of long documents. The problem with extractive summaries of long scientific documents is that, due to length constraints, neural models mostly truncate the input. And this leads to loss of information, especially for extractive models. Therefore, to address this challenge, we consider using a pretrained model (i.e. BERT) to generate local hidden representations between sentences and put them into a graph neural network to learn the complex relations of sentence connections using a self-attention mechanism. For our experiments, we introduce and evaluate the proposed method on two benchmark datasets, namely PubMed and arXiv, on long scientific documents. Experimental results on these two well-known datasets of long scientific documents show promising results of our method.

References

Joshua Ainslie, Santiago Ontañón, Chris Alberti, Philip Pham, Anirudh Ravula, and Sumit K. Sanghai. 2020. Etc: Encoding long and structured data in transformers. *ArXiv*, abs/2004.08483.

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

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2021. Longt5: Efficient text-to-text transformer for long sequences.

Yin Jou Huang and Sadao Kurohashi. 2021. Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3046–3052, Online. Association for Computational Linguistics.

Yaser Keneshloo, Tian Shi, Naren Ramakrishnan, and Chandan K. Reddy. 2018. Deep reinforcement learning for sequence to sequence models.

Bo Pang, Erik Nijkamp, Wojciech Kryściński, Silvio Savarese, Yingbo Zhou, and Caiming Xiong. 2022. Long document summarization with top-down and bottom-up inference.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

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.

Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019a. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, page 793–803, New York, NY, USA. Association for Computing Machinery.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019b. HIBERT: document level pre-training of hierarchical bidirectional transformers for document summarization. *CoRR*, abs/1905.06566.

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.

A Appendix

In this section, we describe the work each member of the group focused on.

- Xiangru Tang: Propose problem to work on, write project proposal, implement the model
- Yizhi Zhao: Write project update, finetuning model on PubMed dataset
- Wenxin Xu: Finetuning model on PubMed dataset, write final paper
- Chengqian Jin: Design model architecture, finetuning model on arXiv dataset