Capturing Relations between Scientific Papers: An Abstractive Model for Related Work Section Generation

Xiuying Chen[123], **Hind Alamro**[34], **Mingzhe Li**[12], **Shen Gao**[1], **Xiangliang Zhang**[3*], **Dongyan Zhao**[125*], **Rui Yan**[6]
[1]Wangxuan Institute of Computer Technology, Peking University, Beijing,China
[2]Center for Data Science, Peking University, Beijing, China
[3]King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
[4]Computer Science Department, Umm Al-Qura University, Makkah, Saudi Arabia
[5]State Key Laboratory of Media Convergence Production Technology and Systems
[6]Gaoling School of Artificial Intelligence, Renmin University of China
xy-chen@pku.edu.cn

Abstract

Given a set of related publications, related work section generation aims to provide researchers with an overview of the specific research area by summarizing these works and introducing them in a logical order. Most of existing related work section generation models follow the inflexible extractive style, which directly extract sentences from multiple original papers to form a related work discussion. Hence, in this paper, we propose a *Relation-aware Related work Generator* (RRG), which generates an abstractive related work section from multiple scientific papers in the same research area. Concretely, we propose a relation-aware multi-document encoder that relates one document to another according to their content dependency in a relation graph. The relation graph and the document representation interact and are refined iteratively, complementing each other in the training process. We also contribute two public datasets composed of related work sections and their corresponding papers[1]. Extensive experiments on the two datasets show that the proposed model brings substantial improvements over several strong baselines. We hope that this work will promote advances in related work section generation task.

1 Introduction

The related work section generation task aims to automatically generate a summary of the most relevant works in a specific research area, which can help researchers to familiarize themselves with the state of the art in the field. Several methods (Hoang and Kan, 2010; Hu and Wan, 2014; Chen and Zhuge, 2019) have been proposed to study how to obtain the related work section automatically by extracting important sentences from multiple original papers. However, extractive approaches lack the sophisticated abilities that are crucial to high-quality summarization such as paraphrasing and generalization, and often lead to a related work section with poor coherence and readability (See et al., 2017; Hsu et al., 2018). For example, as shown in Table 1, the extracted sentences share the pattern "*We find...*" as the subject of sentences, which, as a matter of fact, refer to different authors. On the contrary, the abstractive related work in Table 1 reveals that the works are conducted by different scholars. It also has conjunction words such as "*Furthermore*" and "*However*", which can explain the logical relationship between the cited works, and thus form an elegant narration. Hence, in this paper, we target on the *abstractive related work generation task*, which generates a related work including novel words and phrases not copied from the source text.

Extractive Related Work: *We find that* CRISPR/Cas9 can robustly and specifically reduce the expression of these microRNAs up to 96% [1]. *We find that* miRNA knockdown phenotypes caused by CRISPR/Cas9 transient editing can be stably maintained in both in vitro and in vivo models for a long term (up to 30 days) [2]. Although genome editing using the CRISPR-Cas system is highly efficient in human cell lines, CRISPR-Cas genome editing in primary human cells is more challenging [3].
Abstractive Related Work: Recently, [1] showed that CRISPER-Cas9 targeted miRNA-17, miRNA-200c and miRNA-141, repressed their activity in human colon cancer cell lines HCT116 and HT-29. *Furthermore*, in vivo targeting was effective for at least a month [2]. *However*, off-target mutagenesis and effects of a single miRNA on various gene targets are the limitations to the use of this modern technology specifically in brain disorders like prion diseases [3].

Table 1: Comparison of a related work paragraph generated by an *extractive* method (human-annotated) and an *abstractive* man-made related work paragraph with the same multiple original papers.

* Corresponding author.

[1] https://github.com/iriscxy/relatedworkgeneration

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 6068–6077

There are two main challenges in this task: (1) the related work should summarize the contribution of each paper, and (2) explain the relationship between different papers such as parallel, turning, and progressive relation, so as to introduce them in a logical order. While existing summarization models can address the first problem, they do not target at comparing and explaining the relationship between these articles. Hence, to tackle the above challenges, we propose a *Relation-aware Related work Generator* (RRG), which generates an abstractive related work given multiple scientific papers in the same research area. Firstly, we encode the multiple input articles in a hierarchical manner, obtaining the overall representation for each document. Then, we propose a relation-aware multi-document encoder that relates multiple input documents in a relation graph. In the training process, the relation graph and the document representation interact and are refined iteratively, complementing each other. Finally, in the decoder part, we utilize the relation graph information to assist the decoding process, where the model learns to decide whether to pay attention to the input documents or the relationship between them.

To evaluate our model, we introduce two large-scale related work generation datasets, which are composed of related work sections and their corresponding papers. Extensive experimental results show that RRG outperforms several strong baselines in terms of ROUGE metrics and human evaluations on both datasets.

In summary, our contributions include:

• We address an abstractive related work generation task, which aims to generate an abstractive related work with novel words and phrases.

• We propose a relation-aware multi-document encoder that relates one of the multiple input documents to another, and establishes a relation graph storing the dependency between documents.

• We contribute two public large-scale related work generation datasets that are beneficial for the community.

2 Related Work

We discuss the related work on related work generation and multi-document summarization.

Related Work Generation. Most of the previous related work section generation methods are extractive. For example, Hoang and Kan (2010) take in a set of keywords arranged in a hierarchical fashion to drive the creation of an extractive related work. Later, (Hu and Wan, 2014) first exploits a Probabilistic Latent Semantic Analysis (PLSA) model to split the sentence set of multiple reference papers into different topic-biased parts, and then applies regression models to learn the importance of the sentences. Finally, it employs an optimization framework to generate the related work section. Chen and Zhuge (2019) propose to first construct a minimum Steiner tree of the keywords. Then the summary is generated by extracting the sentences from the papers that cite the reference papers of the paper being written to cover the Steiner tree.

However, abstractive approaches on related work generation have met with limited success. Apart from the lack of sufficient training data, neural models also face the challenge of identifying the logic relationship between multiple input documents.

Multi-document Summarization. The multi-document summarization task aims to cover the key shared relevant information among all the documents while avoiding redundancy (Goldstein et al., 2000). Existing multi-document summarization methods are mostly extractive (Christensen et al., 2013; Parveen and Strube, 2014; Ma et al., 2016; Chu and Liu, 2018). For example, Wang et al. (2020) present a heterogeneous graph-based neural network which contains semantic nodes of different granularity levels apart from sentences. Recently, a vast majority of the literature is dedicated to abstractive multi-document summarization. Lu et al. (2020) propose a large-scale multi-document summarization dataset created from scientific articles. Jin et al. (2020) propose a multi-granularity interaction network for extractive and abstractive approaches. Li et al. (2020a) develop a neural abstractive multi-document summarization model which leverages explicit graph representations of documents to guide the summary generation process.

While the multi-document summarization task aims to extract information shared by multiple documents, related work generation aims to compare and introduce the cited works in logic order.

3 Related Work Generation Dataset

Since there are no public large-scale related work generation datasets, we collect two survey datasets composed of related work sections and their corresponding papers. The first dataset is collected from S2ORC (Lo et al., 2020), which consists of papers in multiple domains (physics, math, computer sci-

Dataset	# Pairs (train/valid/test)	# source (articles)	# words (doc)	# sents (docs)	# words (summary)	# sents (summary)	vocab size
S2ORC	126,655/5,000/5,000	5.02	1,079	45	148	6.69	377,431
Delve	72,927/3,000/3,000	3.69	626	26	181	7.88	190,381
Multi-News	44,972/5,622/5,622	2.78	2,103	82	263	9.97	666,515
RWS	25	9.47	5,496	237	367	18.28	15,019
DUC03+04	320	10	4,636	173	109	2.88	19,734
TAC 2011	176	10	4,695	188	99	1.00	24,672

Table 2: Comparison of our S2ORC and Delve dataset to other related work and multi-document datasets. Training, validation, and testing size splits are provided when applicable. Statistics for multi-document inputs are calculated on the concatenation of all input sources.

ence, *etc.*), and the second is Delve (Akujuobi and Zhang, 2017), which consists of computer science papers. All the papers in each of these two datasets form a large connected citation graph, allowing us to make full use of the citation relationships between papers.

Dataset Preprocessing. For each case, the generation target is a paragraph with more than two citations, as a comprehensive related work usually compares multiple works under the same topic. The abstract of each cited paper is regarded as input, considering that the main idea of a cited paper is described in its abstract. We then conduct a human evaluation to examine the dataset quality. Concretely, we sample 200 cases from both datasets and ask three annotators to state how well they agree with the following statement, on a scale of one to three (disagree, neutral, agree): the related work can be partly generated based on the given abstracts of the cited papers. The evaluation is conducted on the Amazon Mechanical Turk, which has been employed in a variety of NLP tasks including summarization (Liu and Lapata, 2019a), question answering (Gan and Ng, 2019), and dialog system (Li et al., 2020b). The result shows that 94.5% cases win 3 scores, while only 3.5% cases obtain 1 score. This demonstrates the good quality of the datasets.

Statistics. Table 2 compares Delve and S2ORC to other public datasets including DUC data from 2003 and 2004, TAC 2011 data, and Multi-News, which are typically used in multi-document settings. We also list the statistics of a recent related work generation dataset RWS, which is proposed by Chen and Zhuge (2019). The total number of collected samples for the S2ORC and Delve is about 150,000 and 80,000, respectively. It can be seen that Multi-News is most similar to our dataset due to its large-scale. However, the average number of documents per case in Multi-News is smaller than ours.

4 Problem Formulation

Before presenting our approach for related work generation, we first introduce our problem formulation and used notations.

To begin with, for a set of relevant papers $D = (d_1, d_2, \cdots, d_N)$ in a specific area, where d_i denotes a paper, we assume there is a corresponding related work $Y = (y_1, y_2, \cdots, y_T)$. N is the number of relevant papers, and $d_i = (w_1^i, w_2^i, \cdots, w_{N_i}^i)$, where w_j^i is the j-th word in i-th paper, and N_i is the number of words in d_i. T is the number of words in a related work. Given the multiple papers D, our model generates a related work $\hat{Y} = (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{\hat{T}})$. Finally, we use the difference between generated related work $\hat{Y}$ and ground truth related work Y as the training signal to optimize the model parameters.

5 The Proposed RRG Model

5.1 Overview

In this section, we introduce the *Relation-aware Related work Generator* (RRG) in detail. An overview of RRG is shown in Figure 1, which has three main parts:

• *Hierarchical Encoder* reads multiple input documents and learns the multi-level representations for words and documents.

• *Relationship Modeling* relates one paper to another and obtains their relationship graph.

• *Related Work Generator* produces the abstractive related work by attending to the hierarchical representations and the relation graph between documents.

5.2 Hierarchical Encoder

To begin with, each input w_j^i is converted into the vector representation $\hat{e}_j^i$ by the learned embeddings.

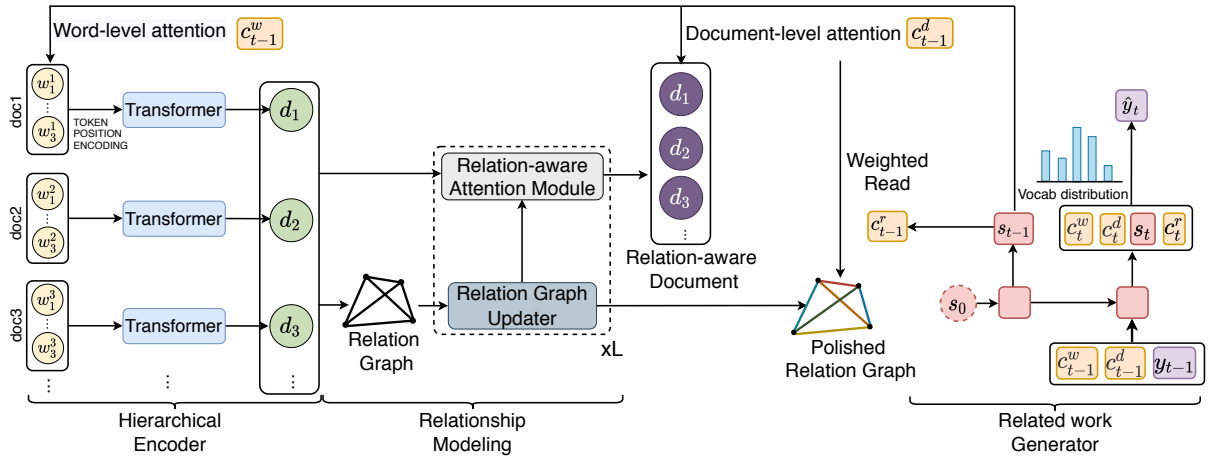

Figure 1: Overview of RRG, which consists of three parts: (1) *Hierarchical Encoder* encodes the multiple inputs in hierarchical levels; (2) *Relationship Modeling* relates one paper to another and stores their relation graph; and (3) *Related Work Generator* generates the related work by attending to input documents and the relationship between them.

We then assign positional encoding (PE) to indicate the position of the word w_j^i where two positions need to be considered, namely document index i and word index j. We concatenate the position embedding PE_i, PE_j to obtain the final position embedding p_j^i. The definition of positional encoding is consistent with the Transfomer (Vaswani et al., 2017). The input word representation e_j^i is obtained by adding embedding $\hat{e}_j^i$ and position embedding p_j^i.

We then perform multi-head self-attention across the word representations in the same document to obtain the contextual word representation $h_{w_j^i}$:

$$h_{w_j^i} = \text{MHAM}(e_j^i, e_*^i), \tag{1}$$

where MHAM denotes the Multi-head Attention Module (Vaswani et al., 2017), and $*$ denotes index $j \in (1, N_i)$. Concretely, The first input is for query and the second input is for keys and values. Each output element, $h_{w_j^i}$, is computed as the weighted sum of linearly transformed input values:

$$h_{w_j^i} = \sum_{l=1}^{N_i} \alpha_{j,l}^i \left(e_l^i W_w^V\right), \tag{2}$$

$$\alpha_{j,l}^i = \frac{\exp\left(\beta_{j,l}^i\right)}{\sum_{k=1}^{N_i} \exp\left(\beta_{j,k}^i\right)}. \tag{3}$$

Here, $\beta_{j,l}^i$ is computed using a compatibility function that compares two input elements:

$$\beta_{j,l}^i = \frac{\left(e_j^i W_w^Q\right)\left(e_l^i W_w^K\right)^T}{\sqrt{d}}, \tag{4}$$

where d is the hidden dimension, and W_w^Q, W_w^K, W_w^V are parameter matrices. From the word-level representation we obtain the overall representation for each document:

$$h_{d_i}^0 = \text{meanpool}\left(\left\{h_{w_1^i}, \cdots, h_{w_{N_i}^i}\right\}\right). \tag{5}$$

5.3 Relationship Modeling

The document representation $h_{d_i}^0$ does not contain cross-document information, thus, it cannot learn richer structural dependencies among textual units. In this subsection, we introduce a novel graph-based *Relationship Modeling* (RM), which not only allows sharing information across multiple documents but also models the logic dependency between documents. Note that it is impossible to explicitly list all the relationships between documents because the relationships vary from document pair to pair depending on the document content, and the content of documents is unlimited. Hence, we model the relationships hidden vectors and let the model capture such diverse relationships by the hidden vectors. Concretely, since the relationship graph is constructed based on the representation of each document, while a comprehensive document representation should consider its relationship with other documents. These two processes complement each other. Hence, our RM module is an iterative module, which has a stack of L identical layers. In each layer, we iteratively update the relationship graph, and then fuse the information from the graph to the document representation, as shown in Figure 2.

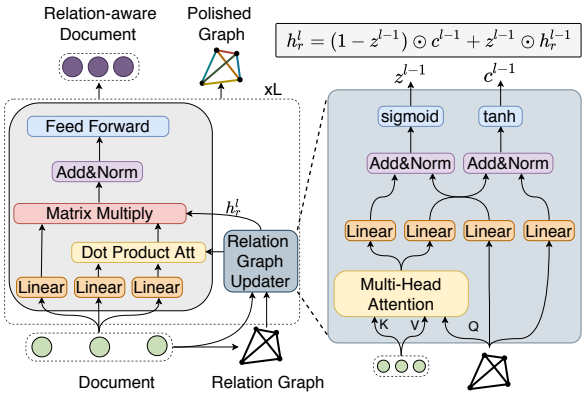

Figure 2: Framework of the relationship modeling, which consists of a relation graph updater (in the right cyan part), and a relation-aware attention module (in the left gray part).

For a start, the relation edge in our graph is initialized by the document representation:

$$h^0_{r_{i,j}} = \text{MLP}_a([h^0_{d_i}; h^0_{d_j}]), \tag{6}$$

where MLP is a multi-layer perceptron, and $[;]$ is the concatenation operation.

In each iteration, we first propose a *Relation Graph Updater* (RGU) to renew the graph based on the polished document representation so far (shown in the right part of Figure 2):

$$h^l_{r_{i,j}} = \text{RGU}(h^{l-1}_{r_{i,j}}, h^{l-1}_{d_*}). \tag{7}$$

Here, $*$ denotes index $i \in (1, N)$, meaning that all document representations will be involved in updating the relation graph. Concretely, RGU first aggregates the information from both the previous graph $h^{l-1}_{r_{i,j}}$ and the document states $h^{l-1}_{d_*}$ from the last layer, using a multi-head attention (MHAM introdced in §5.2). The input for query Q is $h^{l-1}_{r_{i,j}}$, and input for key K and value V is $h^{l-1}_{d_*}$. The output intermediate graph states $s^{l-1}_{i,j}$ are further encoded using a feed-forward layer and then merged with the intermediate hidden states $h^{l-1}_{r_{i,j}}$ using a residual connection and layer norm.

We summarize the procedure below:

$$\begin{aligned}
s^{l-1}_{i,j} &= \text{MHAM}(h^{l-1}_{r_{i,j}}, h^{l-1}_{d_*}),\\
c^{l-1}_{i,j} &= \tanh(W^{l-1}_a h^{l-1}_{r_{i,j}} + W^{l-1}_b s^{l-1}_{i,j}),\\
z^{l-1}_{i,j} &= \text{sigmoid}(W^{l-1}_c h^{l-1}_{r_{i,j}} + W^{l-1}_d s^{l-1}_{i,j}),\\
h^l_{r_{i,j}} &= (1 - z^{l-1}_{i,j}) \odot c^{l-1}_{i,j} + z^{l-1}_{i,j} \odot h^{l-1}_{r_{i,j}},
\end{aligned}$$

where $\odot$ denotes Hadamard product, and $c^{l-1}_{i,j}$ is the internal cell state. $z^{l-1}_{i,j}$ is the update gate that controls which information to retain from the previous memory state. This update strategy is conceptually similar to long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). It differs in that multi-head attention is used and thus multiple graph slots are supported instead of a single one in LSTM, which gives it a higher capacity of modeling complex relations.

Next, the updated graph is fused in the *Relation-aware Attention Module* (RAM) to update the document representation:

$$h^l_{d_i} = \text{RAM}(h^{l-1}_{d_i}, h^{l-1}_{d_*}, h^l_{r_{i,*}}). \tag{8}$$

RAM is similar to MHAM, where $h^{l-1}_{d_i}$ is for query, $h^{l-1}_{d_*}$ is for key and value. However, there are two changes in Equation 2 and Equation 4. Specifically, we modify Equation 2 to propagate edge information to the sub-layer output:

$$h^l_{d_i} = \textstyle\sum_{j=1}^N \alpha^{l-1,r}_{i,j} \left(h^{l-1}_{d_j} W^V_r + h^l_{r_{i,j}} \right). \tag{9}$$

In this way, the representation of each document is more comprehensive, consisting of its relation dependency information with other documents. What is more, when deciding the weight of each edge, *i.e.,* $\beta^{l-1,r}_{i,j}$, we also incorporate relation edge information, since close relationships such as succession or transition can have a great impact on edge weight. Concretely, Equation 4 is changed to:

$$\beta^{l-1,r}_{i,j} = \frac{\left(h^{l-1}_{d_i} W^Q_r \right) \left(h^{l-1}_{d_j} W^K_r + h^l_{r_{i,j}} \right)^T}{\sqrt{d}}. \tag{10}$$

We summarize the whole relationship modeling process as:

$$h^L_d, h^L_r = \text{RM}(h^0_d, h^0_r). \tag{11}$$

For brevity, we omit the subscript L in the following section.

5.4 Related Work Generator

To generate a consistent and informative summary, we propose an RNN-based decoder following (Chen et al., 2019; Gao et al., 2019) that incorporates the outputs of the hierarchical encoder and the relationship graph as illustrated in Figure 1.

Our decoder is a single-layer unidirectional LSTM. At each step t, the decoder updates the hidden state from s_{t-1} to s_t:

$$s_t = \text{LSTM}\left(s_{t-1}, \left[c^w_{t-1}, c^d_{t-1}, e(y_{t-1}) \right] \right). \tag{12}$$

Following previous works (Bahdanau et al., 2015), we employ an attention mechanism to compute the attention distribution over the source words in the sequence-to-sequence structure:

$$\alpha_{t,j}^{w',i} = W_a^g \tanh\left(W_b^g s_t + W_c^g h_{w_j^i}\right), \quad (13)$$

$$\alpha_{t,j}^{w,i} = \exp\left(\alpha_{t,j}^{w',i}\right) / \sum_{l=1}^{N_i} \exp\left(\alpha_{t,l}^{w',i}\right), \quad (14)$$

$$c_t^w = \sum_{i=1}^{N} \sum_{j=1}^{N_i} \alpha_{t,j}^{w,i} h_{w_j^i}, \quad (15)$$

where c_t^w denotes word context vector. Similarly, we extend the attention mechanism to document level:

$$\alpha_{t,i}^{d'} = W_d^g \tanh\left(W_e^g s_t + W_f^g h_{d_i}\right), \quad (16)$$

$$\alpha_{t,i}^{d} = \exp\left(\alpha_{t,i}^{d'}\right) / \sum_{l=1}^{N} \exp\left(\alpha_{t,l}^{d'}\right), \quad (17)$$

$$c_t^d = \sum_{i=1}^{N} \alpha_{t,i}^{d} h_{d_i}. \quad (18)$$

The encoded relationship information is also important for facilitating the transition introduction in the related work, and the specific information in the graph that is needed at each step depends on which document is being introduced. Hence, we employ the document-level attention weights in Equation 17 to read the relationship graph:

$$\begin{aligned} h_{r_i^m} &= \text{meanpool}\left(\left\{h_{r_{i,1}}, \cdots, h_{r_{i,N}}\right\}\right), \\ c_t^r &= \sum_{i=1}^{N} \alpha_{t,i}^{d} h_{r_i^m}. \end{aligned} \quad (19)$$

Finally, an output projection layer is applied to get the final generating distribution P_t^v over vocabulary, as shown in Equation 20:

$$P_t^v = \text{softmax}(\text{MLP}_c[s_t; c_t^w; c_t^d; c_t^r]). \quad (20)$$

Our objective function is the negative log likelihood of the target word y_t:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P_t^v(y_t). \quad (21)$$

In order to handle the out-of-vocabulary (OOV) problem, we equip our decoder with a pointer network (Gu et al., 2016; See et al., 2017). This process is the same as the model described in (See et al., 2017), thus, is omit here due to limited space.

6 Experimental Setup

6.1 Baselines

To evaluate the performance of our proposed model, we compare it with the following baselines:

Extractive Methods:

(1) **LEAD**: selects the first sentence of each document as the summary as a baseline. (2) **TextRank** (Mihalcea and Tarau, 2004): is a multi-document graph-based ranking model. (3) **BertSumEXT** (Liu and Lapata, 2019b): is an extractive summarization model with BERT. (4) **MGSum-*ext*** (Jin et al., 2020): is a multi-granularity interaction network for extractive multi-document summarization.

Abstractive Methods:

(1) **PTGen+Cov**: combines the sequence-to-sequence framework with copy and coverage mechanism in summarization task (See et al., 2017). (2) **TransformerABS**: is an abstractive summarization model based on the Transformer (Vaswani et al., 2017). (3) **BertSumABS** (Liu and Lapata, 2019b): is an abstractive summarization network built on BERT. (4) **MGSum-*abs*** (Jin et al., 2020): is a multi-granularity interaction network for abstractive multi-document summarization. (5) **GS** (Li et al., 2020a): is a neural abstractive multi-document summarization model that leverages well-known graphs to produce abstractive summaries. We use the TF-IDF graph as the input graph.

6.2 Implementation Details

We implement our model in TensorFlow (Abadi et al., 2016) on an NVIDIA GTX 1080 Ti GPU. For all the neural models, we truncate the input articles to 500 tokens in the following way: for each example with S source input documents, we take the first 500/S tokens from each source document. The maximum document number is set to 5. The minimum decoding step is 50, and the maximum step is 100. The word embedding dimension is set to 128 and the number of hidden units is 256. We initialize all of the parameters randomly using a Gaussian distribution. The batch size is set to 16, and we limit the vocabulary size to 50K. We use Adagrad optimizer (Duchi et al., 2010) as our optimizing algorithm. We also apply gradient clipping (Pascanu et al., 2013) with a range of $[-2, 2]$ during training. For the testing, we employ beam search with a beam size of 4 to generate more fluent summaries.

To obtain the extractive oracle, since it is computationally expensive to find a globally optimal subset of sentences that maximizes the ROUGE score, we employ a greedy approach, where we add one sentence at a time incrementally to the

Models	S2ORC Dataset			Delve Dataset		
	RG-1	RG-2	RG-L	RG-1	RG-2	RG-L
oracle ext	38.68	7.23	34.31	38.07	7.21	33.27
Sentence extraction methods						
LEAD	20.60	2.05	16.50	23.18	2.30	19.09
TextRank (Mihalcea and Tarau, 2004)	22.36	2.65	19.73	25.25	3.04	22.14
BertSumEXT (Liu and Lapata, 2019b)	24.62	3.62	21.88	28.43	3.98	24.71
MGSum-*ext* (Jin et al., 2020)	24.10	3.19	20.87	27.85	3.95	24.28
Abstractive methods						
PTGen+Cov (See et al., 2017)	23.54	4.38	21.18	27.54	4.09	24.12
TransformerABS (Vaswani et al., 2017)	21.65	3.64	20.43	26.89	3.92	23.64
BertSumABS (Liu and Lapata, 2019b)	23.63	4.17	21.69	28.02	3.50	24.74
MGSum-*abs* (Jin et al., 2020)	23.94	4.58	21.57	28.13	4.12	24.95
GS (Li et al., 2020a)	23.92	4.51	22.05	28.27	4.36	25.08
RRG	**25.46**	**4.93**	**22.97**	**29.10**	**4.94**	**26.29**
Ablation models						
RRG w/o PP	24.80	4.75	22.30	28.89	4.64	25.60
RRG w/o RM	24.32	4.50	21.95	28.40	4.01	25.12
RRG w/o Upd	24.58	4.71	22.11	28.79	4.13	25.30

Table 3: ROUGE scores comparison between RRG and baselines. All our ROUGE scores have a 95% confidence interval of at most ±0.22 as reported by the official ROUGE script.

summary, such that the ROUGE score of the current set of selected sentences is maximized with respect to the entire gold summary.

7 Experimental Results

7.1 Automatic Evaluation

Following Chen et al. (2018), we evaluate summarization quality using ROUGE F_1 (Lin, 2004). We report unigram and bigram overlap (ROUGE-1 and ROUGE-2) to assess the informativeness and the longest common subsequence (ROUGE-L) as a means of the assessing fluency.

Table 3 summarizes our results. The first block in the table includes extractive systems, and the second block includes abstractive baselines. As can be seen, abstractive models generally outperform extractive ones, especially in terms of ROUGE-L scores. We attribute this result to the observation that the gold related work of this dataset tends to use novel word combinations to summarize the original input documents, which demonstrates the necessity of solving the abstractive related work generation task. Among abstractive models, surprisingly, BertSumABS does not perform as well as other state-of-the-art baselines. This is probably because BERT does not fit well on scholar data that have technical terms. Finally, our model RRG gains an improvement of 1.83 (1.08) points compared with BertSumABS, 1.54 (0.83) points compared with GS on ROUGE-1 on S2ORC (Delve),

	QA(%)	Inform	Coh	Succ
BertSumABS	26.8	1.86	1.93	1.80
MGSum-*abs*	29.9	2.03	1.96	1.90
GS	32.8	2.23	2.06	2.03
RRG	**38.8**	**2.37**	**2.16**	**2.10**

Table 4: Model scores based on questions answered by AMT participants and summary quality rating.

verifying the effectiveness of our RRG.

Table 3 also summarizes ablation studies aiming to assess the contribution of individual components in our RRG model. The results confirm that the encoding paragraph position in addition to token position within each paragraph is beneficial (see row w/o PP), as well as relationship modeling (row w/o RM). Updating the relation graph also helps the summarization process, where removing the update mechanism causes ROUGE-L drop by 0.86 (0.99) (row w/o Upd) on S2ORC (Delve) dataset.

7.2 Human Evaluation

We also assessed the generated results by eliciting human judgments on 30 randomly selected test instances from Delve dataset. Our first evaluation study quantified the degree to which summarization models can retain the key information following a question-answering paradigm (Liu and Lapata, 2019a). We created a set of questions based on the gold-related work and examined whether participants were able to answer these questions by reading generated related works. The principle

GOLD	given a set of annotated images as training data , many methods have been proposed in the literature to find most representative keywords to annotate new images [1] [2] . however , in most cases , the labeled data are insufficient . compared to the large size of an image or video data set , the annotated images have a relative small number . the semisupervised learning techniques leverage the unlabeled data in addition to labeled data to tackle this difficulty [3] [4] .
QA	Are labeled data large enough for image annotation in most cases? [no] What techniques have been proposed to leverage the unlabeled data? [semisupervised techniques]
M-*ext*	we introduce a new method to automatically annotate and retrieve images using a vocabulary of image semantics [1] . we evaluate the innovative sdf-based approach on corel images compared with support vector machine-based approach . in this paper , we propose a novel scheme that exploits both semi-supervised kernel learning and batch mode active learning for relevance feedback in cbir [3] . this paper presents a novel semi-supervised learning method which combines the power of learned similarity functions and classifiers [4] .
M-*abs*	active learning methods have been widely used in computer vision tasks , including speech recognition and computer vision [1] [2] . in the context of machine learning , there has been substantial research effort in the field of active learning [3] . [4] address the problem of active learning to a set of labeled data based on the idea of boosting .
GS	the problem of image annotation has been studied extensively in recent years [1] [2] . in contrast , the majority of work on image annotation has focused on reducing the amount of the number of required training samples , such as semi-supervised learning ([3]) , support vector machine (svm [4]) .
RRG	there has been a lot of work on image annotation learning [1] [2] . however , most of the existing methods is not applicable when training data size is small . [3] propose a semi-supervised learning algorithm for active learning such as local svm. [4] address the problem of active learning to a set of labeled data .

Table 5: Gold human authored summaries, questions based on them (answers shown in square brackets) and automatic summaries produced by MGSum-*ext*, MGSum-*abs*, GS, and our RRG. Blue denotes inconsistent sentences, while pink denotes relation conjunction that explains the relationship between different works.

for writing a question is that the information to be answered is about factual description, and is necessary for a related work section. Two Ph.D. students majoring in computer science (also the authors) wrote five questions independently for each sampled ground truth related work since the Delve dataset also consists of computer science papers. Then they together selected the common questions as the final questions that they both consider to be important. Finally we obtain 67 questions, where correct answers are marked with 1 and 0 otherwise. Examples of questions and their answers are given in Table 5. Our second evaluation study assessed the overall quality of the related works by asking participants to score them by taking into account the following criteria: *Informativeness* (does the related work convey important facts about the topic in question?), *Coherence* (is the related work coherent and grammatical?), and *Succinctness* (does the related work avoid repetition?). The rating score ranges from 1 to 3, with 3 being the best. For both evaluation metrics, a model's score is the average of all scores.

Both evaluations were conducted on the Amazon Mechanical Turk platform with 3 responses per hit. Participants evaluated related works produced by the BertSumABS, MGSum-*abs*, GS, and our RRG. All evaluated models are those who achieved the best performance in automatic evaluations. Table 4 lists the average scores of each model, showing that RRG outperforms other baseline models among all metrics. We calculate the kappa statistics in terms of informativeness, coherence, and succinctness, and the scores are 0.38, 0.29, 0.34, respectively. To verify the significance of these results, we also conduct the paired student t-test between our model and GS (the row with shaded background). We obtain a p-value of 6×10^{-6}, 5×10^{-9}, and 7×10^{-7} for informativeness, coherence, and succinctness. Examples of system output are provided in Table 5. We can see that related work generated by RRG correctly captures the relationship between papers [1,2] and [3,4], and successfully summarizes the contributions of corresponding papers. Among baselines, MGSum-*ext* fails to connect the cited papers in logic. MGSum-*abs* and GS fail to capture the transitional relationship between the first two works and the last two works.

7.3 Analysis of Relation Graph

To fully investigate what is stored by the relation graph, we draw a heatmap of the graph for the case in Table 5. Since the edge in relation graph is a vector containing semantic meaning, which cannot be directly explained, we use the edge between paper [2] and [3] as a benchmark and compute the cosine similarity between the benchmark and other relation edges. Dark color means that the relationship between the corresponding two papers is similar with edge [2]-[3], and vice versa. We already know that there is a transitional relationship between [2] and [3], so if an edge has a high cosine similarity

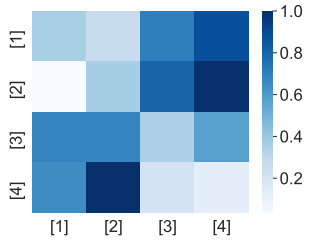

Figure 3: The similarity between each relation edge and paper [3]-[4] edge. The darker the color is, the higher the similarity is.

with [2]-[3] pair, then the two papers on this edge also form a transitional relationship. As shown in Figure 3, relationship vectors between paper [1], [2] with [4] are relatively more similar to [2]-[3] pair. This is consistent with the fact that paper [1] and [2] are parallel with each other, while they form a transitional relationship compared with [4]. Note that the heatmap is not symmetrical because our relation graph is a bipartite graph.

8 Conclusion

In this paper, we conceptualized the abstractive related work generation task as a machine learning problem. We proposed a new model that is able to encode multiple input documents hierarchically and model the latent relations across them in a relation graph. We also come up with two public large-scale related work generation datasets. Experimental results show that our model produces related works that are both fluent and informative, outperforming competitive systems by a wide margin. In the future, we would like to apply our model to abstract generation and paper generation tasks.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Key Research and Development Program of China (No. 2017YFC0804001), the National Science Foundation of China (NSFC No. 61876196 and NSFC No. 61672058). Rui Yan is partially supported as a Young Fellow of Beijing Institute of Artificial Intelligence (BAAI).

Ethics Impact

In this paper, we propose a relation-aware related work generator which aims to provide researchers with an overview of the specific research area by summarizing the related works and introducing them in a logical order. The positive impact lies in that it can help improve the work efficiency of scholars. The negative impact may be that in some extreme cases, the system may not be able to give an accurate and faithful related work, which can be misleading. Hence, in such situation, scholars should not directly employ the generated related work as the final edition. Instead, they can rely on this system to give insightful related work suggestion.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.

Uchenna Akujuobi and X. Zhang. 2017. Delve: A dataset-driven scholarly search and analysis system. *ACM SIGKDD Explorations Newsletter*, 19:36–46.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR*.

Jingqiang Chen and Hai Zhuge. 2019. Automatic generation of related work through summarizing citations. *Concurr. Comput. Pract. Exp.*, 31.

Xiuying Chen, Zhangming Chan, Shen Gao, Meng-Hsuan Yu, Dongyan Zhao, and Rui Yan. 2019. Learning towards abstractive timeline summarization. In *IJCAI*.

Xiuying Chen, Shen Gao, Chongyang Tao, Yan Song, Dongyan Zhao, and Rui Yan. 2018. Iterative document representation learning towards summarization with polishing. In *EMNLP*.

Janara Christensen, Stephen Soderland, Oren Etzioni, et al. 2013. Towards coherent multi-document summarization. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1163–1173.

Eric Chu and Peter J. Liu. 2018. Unsupervised neural multi-document abstractive summarization. *ArXiv*, abs/1810.05739.

John C. Duchi, Elad Hazan, and Yoram Singer. 2010. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159.

Wee Chung Gan and H. T. Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *ACL*.

Shen Gao, Xiuying Chen, Piji Li, Zhangming Chan, Dongyan Zhao, and Rui Yan. 2019. How to write summaries with patterns? learning towards abstractive summarization through prototype editing. In *EMNLP*.

Jade Goldstein, Vibhu O Mittal, Jaime G Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Coling 2010: Posters*, pages 427–435.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141.

Yue Hu and Xiaojun Wan. 2014. Automatic generation of related work sections in scientific papers: an optimization approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1633.

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

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

Yu Li, Kun Qian, Weiyan Shi, and Z. Yu. 2020b. End-to-end trainable non-collaborative dialog system. In *AAAI*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081.

Yang Liu and Mirella Lapata. 2019b. 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*, pages 3721–3731.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. 2020. S2ORC: The Semantic Scholar Open Research Corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-xscience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074.

Shulei Ma, Zhi-Hong Deng, and Yunlun Yang. 2016. An unsupervised multi-document summarization framework based on neural document model. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1514–1523.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Daraksha Parveen and Michael Strube. 2014. Multi-document summarization using bipartite graphs. In *Proceedings of TextGraphs-9: the workshop on Graph-based Methods for Natural Language Processing*, pages 15–24.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR.

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

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 *NIPS*, pages 5998–6008.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuan-Jing 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.