

## Article

# Abstractive Summary of Public Opinion News Based on Element Graph Attention

Yuxin Huang <sup>1,2</sup>, Shukai Hou <sup>1,2</sup>, Gang Li <sup>1,2</sup> and Zhengtao Yu <sup>1,2,\*</sup>

<sup>1</sup> Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

<sup>2</sup> Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, China

\* Correspondence: ztyu@hotmail.com

**Abstract:** The summary of case–public opinion refers to the generation of case-related sentences from public opinion information related to judicial cases. Case–public opinion news refers to the judicial cases (intentional homicide, rape, etc.) that cause large public opinion. The public opinion news in these cases usually contains case element information such as the suspect, victim, time, place, process, and sentencing of the case. In the multi-document summary of case–public opinion, due to the problem of information cross and information redundancy between different documents under the same case, in order to generate a concise and smooth summary, this paper proposes an abstractive summary model of case–public opinion based on the attention of a case element diagram. Firstly, multiple public opinion documents in the same case are split into paragraphs, and then the paragraphs and case elements are coded based on the transformer method to construct a heterogeneous graph containing paragraph nodes and case element nodes. Finally, in the decoding process, the two-layer attention mechanism is applied to the case element node and paragraph node, so that the model can effectively solve the redundancy problem in summary generation.

**Keywords:** summary of public opinion of the case; multi-document summaries; case elements; graph attention mechanism



**Citation:** Huang, Y.; Hou, S.; Li, G.; Yu, Z. Abstractive Summary of Public Opinion News Based on Element Graph Attention. *Information* **2023**, *14*, 97. <https://doi.org/10.3390/info14020097>

Academic Editor: Arkaitz Zubiaga

Received: 11 December 2022

Revised: 29 January 2023

Accepted: 3 February 2023

Published: 6 February 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Public opinion related to a case is rapidly fermented and disseminated through the Internet, and a large amount of public opinion information about the case will be formed. From this public opinion information, a brief summary of the topic information is generated, which plays an important role in quickly understanding the case and grasping the trend of public opinion.

In recent years, sequence-to-sequence-based models have achieved good results in single-document summarization tasks, but when applied to multi-document summarization tasks, due to too many input sequences, significant information related to summarization cannot be extracted, causing the generated summary to contain redundant information.

It is difficult to achieve better results simply by applying the sequence-to-sequence model to the multi-document summarization task. The multi-document summary text of case–public opinion usually contains information such as “victims, criminal suspects, and the location of the crime”. This information is an important part of the case–public opinion text and is also important for summary generation, as shown in Table 1.

**Table 1.** Examples of public opinion text data for multi-document cases.

<p><b>Exemplar 1:</b> The reporter learned from the Anti-gangland Office of Hunan Province and the Huaihua Municipal Party Committee that the historical backlog of <b>Xinhuang’s “playground burial case” (the case of Deng Shiping’s murder)</b>, has been thoroughly investigated. <b>Du Shaoping</b> and his accomplice <b>Luo Guangzhong</b> were arrested according to law and prosecuted on suspicion of intentional homicide: <b>Huang Bingsong</b> and other 19 public officials involved in the case received corresponding party affiliation and government sanctions such as expulsion from the party and public office.</p> <p><b>Exemplar 2:</b> The First Instance of intermediate People’s Court of Huaihua City, Hunan Province held a public hearing of the defendant <b>Du Shaoping’s</b> intentional homicide case and the case of a vicious criminal group and pronounced the verdict in court. <b>Deng Shiping</b> and <b>Yao Benying</b> (deceased) from the General Affairs Office of <b>Xinhuang No. 1 Middle School</b> supervised the quality of the project. During the construction process, <b>Du Shaoping</b> had conflicts with <b>Deng Shiping</b> due to issues such as project quality, and held a grudge against <b>Deng Shiping</b>. On January 22, 2003, Du Shaoping and Luo Guangzhong killed Deng Shiping in the office of the engineering headquarters.</p> <p><b>Exemplar 3:</b> The Huaihua Intermediate People’s Court held that defendant <b>Du Shaoping</b>, together with defendant <b>Luo Guangzhong</b>, deliberately and illegally deprived others of their lives, resulting in the death of one person; intentionally injured another person’s body, resulting in minor injuries to one person; organized and led a criminal group of evil forces to carry out quarrels, provocation, illegal detention etc. Defendant <b>Luo Guangzhong</b> was convicted of intentional homicide and sentenced to death with a two-year reprieve and deprivation of political rights for life.</p>
<p><b>Case Elements:</b></p> <p>Element Name: element information</p> <p>Case Name: Xinhuang’s “playground burial case” (the case of Deng Shiping’s murder)</p> <p>Victim: Deng Shiping</p> <p>Suspect: Du Shaoping, Luo Guangzhong, Huang Bingsong</p> <p>Burial Site: Xinhuang No. 1 Middle School Stadium</p> <p>The time of the incident: January 22, 2003</p> <p>Victim found time: June 20, 2019</p>
<p><b>Summary:</b> In the early morning of <b>June 20, 2019</b>, the Public Security Bureau of Xinhuang County, Hunan Province dug up a body in the runway of <b>Xinhuang No. 1 Middle School</b>, and found out a murder case that happened 16 years ago. On December 30, the then-principal Huang Bingsong was sentenced to 15 years in prison for the “<b>playground burial case</b>”. On <b>January 20</b>, the Intermediate People’s Court of Huaihua City, Hunan Province executed <b>Du Shaoping</b> in accordance with the law. In June 2020, <b>Deng Shiping</b> was found to be injured at work and received a subsidy of 880,000 yuan, and his family gave up civil compensation.</p>

As shown in Table 1, in the three different body texts of the “SAN Huang Playground burial Case (Murder case of Tang Shi ping) “Deng Shipping, Du Shaoping, Huang Bingsong, Xinhuang’s “playground burial case””, case elements such as these appear frequently in texts and summaries. We believe that sentences containing case elements are more likely to become abstract sentences, and with the help of case elements, the relationship between documents can be efficiently encoded, and the problems of sentence salience and information redundancy can be resolved.

Based on the above analysis, we build a heterogeneous graph composed of document paragraph nodes and case element nodes. The element nodes can connect different documents and model relationships between documents through case elements. The graph attention network (GAT) [1] is applied to implement the information flow between nodes and iteratively update the node representation. In the decoding process, the decoder first processes the case element node information and expresses the attention weight of the document by combining the attention weight of the case element with the edge weight. We design a novel two-layer attention mechanism to identify salient information in each decoding step by considering the element and the global interaction between documents in the graph and re-addressing the redundancy problem of abstractive summaries. Experi-

ments show that the model significantly improves the performance of case element-based multi-document summarization.

## 2. Related Technologies

### 2.1. Abstractive Summarization Methods

Abstractive summarization methods refer to understanding and summarizing the core ideas of input documents and then generating summaries. The summaries generated by this method are composed of new sentences and have a high semantic fit with the original text. Traditional abstractive summarization methods can be divided into sentence fusion [2–4], topic paraphrasing [5,6], and methods based on information extraction [7–9]. With the in-depth research into deep learning, abstractive methods have achieved good results for single-document summarization [10–12]. Among them, transformer-based methods have become the mainstream methods for abstractive summarization and pre-training language models. Zhang et al. [13] proposed a Hierarchical Bidirectional Encoder Representations from Transformers (HIBERT) model and constructed two types of BERT for document encoding and pre-training with unlabeled data. From the pre-trained encoder, a method to initialize the model to classify sentences is proposed. In addition, several general sequence-to-sequence pre-training models are proposed, such as T5 [14] and BART [15], which are further fine-tuned for summarization tasks to optimize the pre-trained models. Zhang et al. [16] proposed the PEGASUS model and designed a self-supervised pre-training model specifically for abstractive summarization, removing or masking a key sentence from the document and generating this key sentence according to other sentences in the document; these sentences, just as a summary sentence, are similar to an extractive summary. Zou et al. [17] proposed pre-training an abstractive summarization model based on sequence-to-sequence on unlabeled text. By pre-training the model to recover the source text for generating summaries given artificially constructed input text, three sequence-to-sequence pre-training target models are proposed, which include sentence rearrangement, next sentence generation, and masking document generation; these three goals are closely related to abstractive summarization tasks. Another solution that can be borrowed from SDS is to use a multi-layer transformer architecture to scale the length of documents allowing pre-trained LMs to encode a small block of text, and the information can be shared among the blocks between two successive layers [18].

At present, the abstractive summarization method is composed of new sentences, which is highly consistent with the semantic and thematic information of the original text. Therefore, this method has become the mainstream research method.

### 2.2. Graph-Based Summarization Methods

Graph-based summarization methods include traditional graph-based methods and graph-based neural network methods.

The traditional graph-based summarization method refers to sorting the text units on the graph and then selecting the text units with significant information to form the summarization. LexRank [19] calculates sentence saliency scores by connecting feature vector centers in the graph by cosine similarity and then extracts sentences with high scores to form summaries. Wan et al. [20] proposed a multi-document summarization model based on a graph sorting algorithm, which combines document-level information and sentence-to-document relationships and applies it to the graph-based sorting process. Christensen et al. [21] proposed a joint model to select and sort based on indicators, including discourse cues, verb nouns, references, etc., to construct a multi-document graph to represent the discourse relationship between sentences and to estimate the value of a candidate abstract.

In the method of graph neural network, Tan et al. [22] introduced a graph-based attention mechanism in the traditional encoder–decoder model to identify salient sentences; on the decoder, they proposed a hierarchical decoding model, which is a reference mechanism introduced to improve the novelty, information correctness, and fluency of summaries. Yasunaga et al. [23] constructed an approximate discourse graph based on discourse markers and case element links and then applied a graph convolutional network on the relational graph to

score sentences. Fan et al. [24] proposed a query-based model for open-domain natural language processing tasks, building a local graph knowledge base, compressing network search information, reducing redundancy, and then linearizing it into a structured input sequence. Models can encode graph representations in a standard sequence-to-sequence setting. Huang et al. [25] further designed a graph encoder and improved the graph attention network using a dual encoder, document encoder, and graph encoder to maintain the contextual global and local informative features of entity information. Wang et al. [26] constructed heterogeneous graphs by introducing text nodes of different granularity levels to achieve the task of extractive summarization. These text nodes act as intermediaries between sentences and enrich the relationship of cross sentences.

The method of Wang et al. [26] is to construct the correlation between sentences by introducing word nodes. However, when the article is too long, or there are many documents, the graph structure constructed with sentences as nodes will contain a lot of node information, which increases the complexity of graph structure and calculation and consumes a lot of computing resources in model training. Based on this, we connect the paragraph nodes through the case elements. Due to the limited case elements, the constructed element relation graph can effectively reduce the complexity of the graph structure and reduce the computational resources. Song et al. [27] first utilize an off-the-shelf constituency parser to obtain the constituency tree for each sentence. Then, we propose a generic syntax-aware heterogeneous graph attention network to learn the representation for each type of node in this constructed tree-like graph.

### 2.3. Multi-Document Summarization Methods

Multi-document summarization (MDS) is an efficient information aggregation tool that generates informative and concise summaries from clusters of documents related to a topic. Deep learning algorithms learn salient features of sentences or documents through backpropagation to minimize a given objective function.

The encoder–decoder structure is a commonly used model paradigm, where the encoder embeds source documents into hidden representations to generate word, sentence, and document representations. This representation contains compressed semantic and syntactic information, which is then passed to the decoder, which processes the underlying embeddings and synthesizes local and global semantic/syntactic information to produce summaries. For example, Jin et al. [28] proposed a transformer-based multi-granularity interaction network MGSum to unify extractive and generative multi-document summarization. Words, sentences, and documents are considered as three granular semantic units connected by granular hierarchical relational graphs. At the same granularity, a self-attention mechanism is used to capture semantic relations. Sentence-grained representations are used in extractive summarization, word-granular representations are used in abstractive summarization, and a fusion gate is used to integrate and update semantic representations. In addition, an alternate attention mechanism is used to ensure that the summarizer focuses on important information.

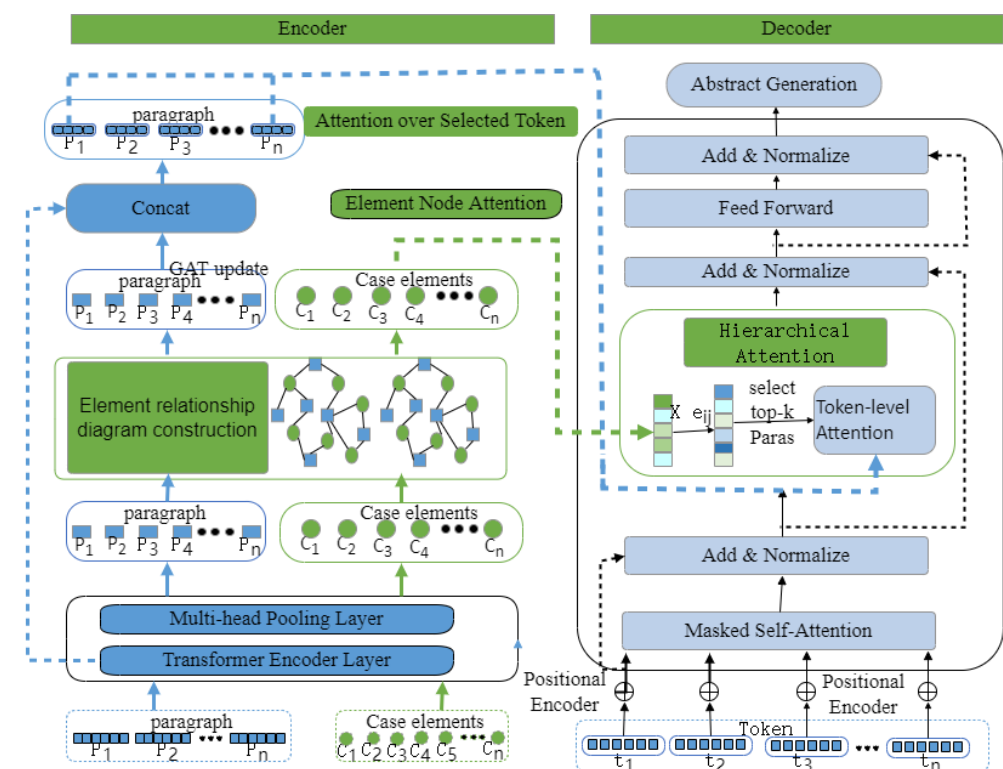
There are also methods that take multiple document connections as input into a neural network model to capture the underlying representation. Another deep neural network model is cascaded to generate a high-level representation based on the previous model. Hierarchical networks enable models to capture abstract and semantic-level features more precisely. For example, Yang et al. [29] proposed a two-stage hierarchical transformer model with inter-paragraph and graph information attention mechanisms, allowing the model to hierarchically encode multiple input documents. A logistic regression model is used to select top-K passages and feed them into a local transformer layer to obtain contextual features. A global transformer layer mixes contextual information to model dependencies of selected passages.

The pre-trained language model is one of the most commonly used methods; the transformer pre-trained on a large text corpus can be fine-tuned by the end-to-end method of the decoder for migration learning, which has achieved the purpose of helping the model

training. Because pre-trained language models can be trained on non-summarized or SDS datasets, the problem of lack of data for multi-document summarization can be overcome. Using a multi-layer transformer architecture to scale the length of the document allows a pre-trained language model to encode a small chunk of text, and information can be shared between chunks of two consecutive layers. BART [15], GPT-2 [30], and T5 [14] are pre-trained language models that can be used for language generation, and they have been applied to multi-document summarization tasks. Compared with conventional pre-trained language models, PEGASUS [16] is a pre-trained language model based on transformer encoder and decoder structure with gap sentence generation (GSG) focusing on summary generation. GSG shows that importance-based sentence masking, not by random or guided selection, is better suited for downstream summarization tasks.

### 3. Materials and Methods

A multi-document abstractive summarization model for case–public opinion based on feature graph attention is based on a transformer-based encoder–decoder architecture [31]. Using the graph neural network as the encoder, combined with the case element information and graph structure information representation, multiple documents can be effectively encoded; in the decoding process, a new two-level attention mechanism is proposed to deal with the saliency and redundancy problem. The specific structure is shown in Figure 1.



**Figure 1.** Multi-document abstractive summarization method for case–public opinion based on feature graph attention.

As shown in Figure 1, the encoder–decoder framework is followed in the model graph, the encoder side is the document paragraph encoder and the case feature encoder, and the decoder is a two-level attention mechanism.

#### 3.1. Element Relationship Diagram Construction Module

The input source documents are multiple documents  $D = \{d_1, d_2, \dots, d_n\}$ , which are first divided into smaller semantic unit paragraphs  $p = \{p_1, p_2, \dots, p_n\}$ . Then they are constructed into a heterogeneous graph  $G = (V, E)$ .  $V$  includes paragraph nodes  $V_p$  and

case feature nodes  $V_c$ .  $E$  represents an undirected edge between nodes; there is no edge inside a paragraph node or case element node, there is only an edge between paragraph nodes or case element nodes. The edges of  $P_i$  and  $C_j$  indicate that the case elements in  $C_j$  are contained in  $P_i$ .

In order to include more information in the diagram, the composition obtains the matrix  $E \in R^{m \times n}$ , where  $e_{ij} \neq 0$  indicates that the elements  $C_j$  of the case are contained in  $P_i$ . Specific algorithm construction is shown in Algorithm 1.

---

**Algorithm 1:** Algorithm for constructing element relation graph

---

Input: input text  $\mathbf{D} = \{d_1, d_2, \dots, d_n\}$

Output: element relationship diagram  $G = \{V, E\}$ ,  $V = C \cup P$

1. Collection of case data sets  $C$ ;
  2. With the document paragraph node as the initial node, to  $V = P$ ;
  3. For  $d_i$  in  $\mathbf{D}$  do;
  4. for  $p_i$  in  $d_i$  do;
  5. if  $p_i$  contains  $c \in C$  then;
  6.  $V = p_i \cup c$ ;
  7. end if;
  8. End for;
  9. End for;
  10. For  $d_i$  in  $\mathbf{D}$  do;
  11. for  $p_i$  in  $d_i$  do;
  12. if  $p_i$  and  $p_j$  contains  $c \in C$  then;
  13.  $E = e_{ij}$ ;
  14. end if;
  15. end for;
  16. End for.
- 

### 3.2. Document Encoder

Segment multiple documents and then stack several token-level transformer [31] encoding layers to encode the contextual information in each paragraph.

$$h_w^l = \text{LayNorm}(x_w^{l-1} + \text{MHAttn}(x_w^{l-1})) \quad (1)$$

$$x_w^l = \text{LayerNorm}(h_w^l + \text{FFN}(h_w^l)) \quad (2)$$

$$h_p = \text{MHPool}(h_{w1}, h_{w2}, \dots) \quad (3)$$

### 3.3. Graph Encoder

The representation of semantic nodes is updated using graph attention networks (GAT) 1. Use  $i, j \in \{1, 2, \dots, (m+n)\}$  to represent any node in the graph, and the adjacent node set of the  $i$  node is represented by  $N_i$ . The GAT layer is designed as follows:

$$z_{ij} = \text{LeakyReLU}(\mathbf{W}_a[\mathbf{W}_q h_i; \mathbf{W}_k h_j]) \quad (4)$$

$$\tilde{z}_{ij} = \tilde{e}_{ij} \times z_{ij} \quad (5)$$

$$\alpha_{ij} = \frac{\exp(\tilde{z}_{ij})}{\sum_{l \in N_i} \exp(\tilde{z}_{il})} \quad (6)$$

$$u_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \mathbf{W}_v h_j\right) \quad (7)$$

where  $\tilde{e}_{ij}$  are edge weights derived from a matrix of TFIDF values. The main idea is to represent edge weights by discretizing real values into integers and then learn the embedding of these integers to map the weights to a multidimensional embedding space  $e_{ij} \in R^{d_e}$ . The TFIDF value indicates the proximity between the case element node and the



paragraph node. Therefore, we directly incorporate the original TFIDF information into the GAT mechanism by updating the attention weights using Equation (5).

Combining GAT and multi-head operations obtains  $h_i$ , adding a residual connection to avoid vanishing gradients after a few iterations.

$$\tilde{h}_i = h_i + u_i \quad (8)$$

We use the above GAT layer and a position feed forward layer to iteratively update the node representation. Each iteration consists of a paragraph-to-case element and case-element-to-paragraph update process. After  $t$  iterations, we represent each input feature matrix with  $\tilde{H}_{pc}$ .

$$\tilde{H}_{pc} \in R^{n_c \times (d_c + d_h)} \quad (9)$$

### 3.4. Element Decoder Based on Two-Layer Attention

Under the multi-document summarization task, the input source document may contain a lot of tokens. If the decoder computes attention weights for all tokens, the cost will be very high, and attention may be distracted. Therefore, this paper proposes that the two-level decoding process first focuses on case element nodes, which can be regarded as saliency indicators in the summarization process. This metric restricts token-level attention to certain passages, which further reduces redundancy compared to the focus on all tokens approach.  $i$  and  $j$  are used to denote case element nodes and paragraph nodes, respectively.

At each decoding step, the state of the decoder is  $s$ , and we compute the attention score of the case element node  $c_i$ .

$$z_i = u_0^T \text{LeakyReLU}([\mathbf{W}_q s; \mathbf{W}_k c_i]) \quad (10)$$

In Equation (10),  $u_0^T$  denotes the transposed matrix obtained after training.

$$\tilde{z}_j = \sum_{i=1}^m z_i \times \tilde{e}_{ij} \quad (11)$$

In Equation (11),  $\tilde{e}_{ij}$  is the edge weight derived from the TFIDF value matrix, and  $\tilde{z}_j$  represents the segment node coefficient to realize the information flow between the element node and the segment node.

$$\beta_j = \frac{\exp(\tilde{z}_j)}{\sum_{l=1}^m \exp(\tilde{z}_l)} \quad (12)$$

In Equation (12), through the normalization operation, weighted summation after an activation function obtains the attention weight of the element node. We select the previous paragraph node with the highest attention score  $\beta_j$ , and then use the attention mechanism for  $T_w$  tokens in the selected paragraph node.

$$z_{wi} = u_1^T \text{LeakyReLU}([\mathbf{W}_q s; \mathbf{W}_k \tilde{h}_{wi}]) \quad (13)$$

In Equation (13),  $u_1^T$  denotes the transition matrix obtained after training.  $z_{wi}$  is the attention coefficient of the node token level in the state  $s$ .  $\tilde{h}_{wi}$  represents the context vector at the token level obtained by segmenting the paragraph.

$$\gamma_{wi} = \frac{\exp(z_{wi})}{\sum_{l=1}^{T_w} \exp(z_{wi})} \quad (14)$$

Equation (14) obtains the attention weight at the token level in the paragraph node through the normalization operation, weighted summation, and an activation function.

$$\hat{\gamma}_{wi} = \beta_j \times \gamma_{wi} \quad (15)$$

In Equation (15), the token-level segment feature of two-level attention mechanism is obtained by combining the element node attention  $\beta_j$  with the token-level segment feature aggregation.

$$v_t = \sum_t \hat{\gamma}_{wi} \hat{h}_{wi} \quad (16)$$

The context vector is taken as the prominent content generated from the summary of the source document and connected with the hidden state  $s_t$  of the decoder to obtain the distribution of the vocabulary.

$$P_{vocab} = \text{Softmax}(\mathbf{W}_o[s_t; v_t]) \quad (17)$$

### 3.5. Parameter Training

The training process follows a traditional sequence-to-sequence model with maximum likelihood estimation as the loss function:

$$L_{seq} = -\frac{1}{|D|} \sum_{(y,x) \in D} \log p(y|x; \theta) \quad (18)$$

where  $x$  and  $y$  are the document–summary pairs from the training set  $D$  and  $\theta$  which are the parameters to learn.

## 4. Results

### 4.1. Case–Public Opinion Multi-Document Summary Dataset


Due to the continuous generation and rapid accumulation of online public opinion information, a large amount of public opinion information related to each case has been generated on the network platform. The case–public opinion multi-document summary data set constructed in this paper is collected from the Internet based on crawler technology. The specific methods are as follows:

First, perform a search in the Sogou Encyclopedia by case category (intentional homicide, robbery, kidnapping, drug trafficking, etc.), and use “intentional homicide” as an example; the search results are shown in Figure 2.

Use Scrapy technology to find corresponding case names, remove non-case data, and form a case database containing ten types of case names. Use the case names in the case database to search in Baidu, and use the crawler technology to crawl the Baidu Encyclopedia case names and corresponding links. Taking the “Xinhuang Playground Burial Case” as an example, the search results are shown in Figure 3.

Figure 3 is based on Baidu Encyclopedia’s display of the search results of “Xinhuang Playground Burial Case”. In the figure, the content in box 1 is defined as the standard abstract, and the content in box 2 is defined as the case elements. The content of the text comes from different web page links at the bottom of the Baidu Encyclopedia. Crawl content from different web links describing the same case forms a multi-document dataset. The main collected information includes release time, source, title, and text. After the link is opened, its structure is shown in Figure 4.




 搜狗搜索

Web page WeChat Know almost picture video the medical Chinese translate

Sogou has found about 45,561 related results for you ▽ All time

[Intentional Homicide Case of Wang Moujia -China Court Network](#)  
 China Court Network March 2, 2021  
 The people's court sentenced the defendant Wang Moujia to the crime of intentional homicide and approved the execution of the death penalty. This case was caused by parents who could not correctly handle the conflicts between the minor children and their classmates during school, and broke into the campus classroom with a murder weapon...


[Intentional homicide occurred in Liaocheng, Shandong Province. A 43-year-old woman was killed and the suspect was arrested after absconding](#)  
 Netease September 10, 2022  
 On September 9, 2022, a case of intentional homicide occurred in Liaocheng, Shandong : ...



[Police in Luliang, Shanxi reported an intentional homicide case : 3 dead and 2 injured, the suspect was married to his ex-wife's family...](#)  
 Interface News 2023-1-22  
 On February 5, the Luliang Public Security WeChat public account reported an intentional homicide case : After preliminary investigation, the suspect Li Moumou was emotionally agitated due to family and marital disputes with his ex-wife. Stabbed to death, ex-wife herself and...

[Liao Moumou Intentional Homicide Case -China Court Network](#)  
 China Court Network May 28, 2014  
 Defendant Liao Moumou knew that the behavior of abandoning the newborn baby in the shower room might lead to the death of the baby, but he still let it go, and eventually caused the death of his own baby. His behavior constituted the crime of intentional homicide . The public prosecution accused Liao...

Figure 2. The search results of “intentional homicide” from the Sogou Encyclopedia.

 百度

★ collect 197 189

In the early morning of June 20, 2019, the Public Security Bureau of Xinhuang County, Hunan Province dug up a body in the runway of a school in Xinhuang County, and found out a murder case 16 years ago . On November 26, it was learned from the Hunan Provincial Anti-crime Office and the Hualhua Municipal Party Committee that 19 public officials involved in the case were severely dealt with in accordance with discipline and law <sup>[1-2]</sup> . From December 17th to 18th, the “playground burial case” was sentenced in the first instance, and Du Shaoping was sentenced to death for intentional homicide and other crimes <sup>[3]</sup> . On December 30, Huang Bingsong , the principal at the time of the “playground burial case”, was sentenced to 15 years in prison. <sup>[4]</sup>

On January 10, 2020, the verdict of the second Instance of the “Case of Burying Corpses in the Playground” was pronounced and the original verdict was upheld <sup>[5]</sup> . On January 20, the Intermediate People's Court of Hualhua City, Hunan Province executed Du Shaoping according to law. On April 3, the second-instance trial of the crime of dereliction of duty by public officials related to the “Case of Burying Corpses in the Playground” upheld the original verdict <sup>[6]</sup> . In June, Deng Shiping was found to have suffered a work-related injury and received a subsidy of 880,000 yuan, and his family members gave up civil compensation <sup>[7]</sup> .

On March 26, 2021, “Sweeping Darkness and Evil-For the Peace of the Country ” will be broadcast. The first episode of the feature film used a long time to tell the story behind the handling of the “playground burial case” in Xinhuang, Hunan. Deng Ling , the daughter of the victim Deng Shiping, appeared on the scene for the first time and recalled her writing to the 16th Supervision Group of the Central Anti-criminal Eradication, Until I saw my father's body being dug up on the playground <sup>[35]</sup> .

Chinese name	Sin Huang playground burial case	the suspect	Du Shaoping , Luo Guangzhong , Huang Bingsong
Foreign name	Case of burying corpses in the playground of Xinhuang	Burial site	Xinhuang No.1 Middle School Stadium
	ng No.1 Middle School	Incident time	January 22, 2003
victim	Deng Shiping	When the vic...	June 20, 2019

Figure 3. Baidu's search results for “Xinhuang Playground Burial Case”. In Box 1 is the case description, and in box 2 is the key element of the case.

### Hunan digs and thoroughly investigates Xinhuang's "playground burial case" and 19 public officials involved in the case are severely dealt with in accordance with discipline and law

Source: Xinhua News Agency Release time: 2019-11-26 16:02 share +

The reporter learned from the Hunan Provincial Anti-crime Office and the Huailu Municipal Party Committee that the long-standing historical case Xinhuang's "Case of Burial in the Playground" (Deng Shiping's murder case) dug up in the special campaign against gangsters has been thoroughly investigated. Du Shaoping and his accomplice Luo Guangzhong was arrested in accordance with the law and prosecuted on suspicion of intentional homicide; Huang Bingsong and other 19 public officials involved in the case were respectively expelled from the party and dismissed from public office. 10 of them were arrested in accordance with the law and transferred for review Prosecution; 13 members of Du Shaoping's criminal gang were arrested and prosecuted according to law.

In mid-April 2019, the Xinhuang County Public Security Bureau seized Du Shaoping's criminal gang in the special campaign against gangsters and solicited clues from the society. In May, the Huailu City Public Security Bureau found that Du Shaoping was closely related to the case when it was checking the clues of Deng Shiping's murder assigned by the 16th Supervision Group of the Central Anti-crime and Elimination Group. After in-depth investigation, the public security organs dug up a human remains in the playground of Xinhuang No. 1 Middle School in the early morning of June 20, based on the confessions of Du Shaoping and Luo Guangzhong and the on-site identification, which was confirmed to be the remains of Deng Shiping through DNA identification.

### The second instance of the "playground burial case" related to the crime of malfeasance by public officials upheld the original verdict

Huailu Intermediate People's Court 2020-04-03 11:54 1.625 million page views

On April 3, 2020, the Intermediate People's Court of Huailu City, Hunan Province adopted a remote video sentencing method for the second-instance verdict on the criminal malfeasance of public officials related to Xinhuang's "playground burial case", and ruled to reject the appeals of nine appellants including Huang Bingsong and Yang Xuewen. On appeal, the original judgment was upheld in the whole case.



**Figure 4.** Schematic diagram of the secondary page structure.

Use crawler programs to collect public opinion news about relevant cases based on the Baidu Encyclopedia. Perform manual calibration, cleaning, delete non-case data, and remove noise data such as “\n”. Finally, a case–public opinion summary dataset is constructed. The dataset contains 4569 texts and 13,133 sentences, and the average length of reference abstracts is 190.94. The statistical results of the dataset are shown in Table 2.

**Table 2.** Multi-document summary dataset information.

	Number of Documents	Number of Sentences	Average Sentence Length	Length of Summarization
training sets	3969	50.78	1255	192.21
validation set	300	48.25	1122	190.88
testing set	300	47.66	1107	189.07

#### 4.2. Experimental Parameter Settings

This experiment is carried out on the multi-document summary dataset of case–public opinion, and the ROUGE [32] value is used to automatically evaluate the summary quality. ROUGE-1 (RG-1), ROUGE-2 (RG-2), and ROUGE-L (RG-L) are used as the evaluation indices. The number of transformer encoding layers in the hyperparameters is set to 6, the hidden size is set to 256, the number of heads is set to 8, and the hidden size of the feedforward layer is set to 1024. We truncate the length of the input paragraph and case elements to 100 and 10 tokens, respectively. In the multi-head pooling layer, the number of heads is eight. In the graph encoding process, each layer has 8 headers, and the hidden size is 256. Other training parameters are shown in Table 3.

**Table 3.** Model training parameter settings.

Parameter Name	Parameter Value
training steps	200,000
beam size	5
learning rate	0.002
warm-up	20,000
hyper parameter	$\beta_1 = 0.9, \beta_2 = 0.998$

#### 4.3. Baseline Model Settings

To verify the effectiveness of the proposed method, we choose to compare with transformer and graph-based summarization models:

(1) FT (flat transformer) is a six-layer based encoder–decoder model. The title of the case–public opinion data and the document are connected into a long text, and the first 800 tokens are intercepted as the model input.

(2) T-DMCA (transformer decoder with memory compressed attention model) [33] is based on the cross-attention mechanism of transformer coding, using a transformer decoder and applying a convolutional layer to compress in the self-attention mechanism key and value values.

(3) HT (hierarchical transformer) [29] can efficiently process multiple input documents and extract transformer architectures with the ability to encode documents in a hierarchical manner. Cross-document relationships are first represented by an attention mechanism.

(4) GraphSum [34] uses graphs to represent multi-document generative summarization models of documents and constructs topic relation graphs and chapter structure graphs, and the model uses graphs to encode documents in order to capture the relationship between documents.

## 5. Discussion

### 5.1. Analysis of Experimental Results

The first set of experiments is a comparison experiment between the model in this paper and the four baseline models on the single-document and multi-document summary datasets of case–public opinion. The results are shown in Table 4.

**Table 4.** Baseline model comparison experiment.

Model	RG-1	RG-2	RG-L
FT	30.28	14.12	26.78
T-DMCA	31.22	15.22	26.94
HT	31.94	15.76	26.57
GraphSum	32.52	15.96	26.40
<b>Our Model</b>	<b>32.81</b>	<b>16.78</b>	<b>27.19</b>

It can be seen from the experimental results in the table above: **(1) Compared with the FT model**, the values of **RG-1 and RG-2** are increased by **2.53 and 2.66**, respectively. This is because in the multi-document summary of the FT model, the first 800 words intercepted by this method may cause key information about public opinion of the case to be removed and cannot fully summarize the theme of the article. **(2) Compared with the T-DMCA model**, the values of **RG-1 and RG-2** are improved by **1.59 and 1.56**, respectively, because in the T-DMCA model, the information generated by the multi-layer decoder in the transformer is redundant. In addition, the multi-layer decoder will cause the problem of inefficiency in inference, and parallel matrix operations can improve the decoding speed of the decoder. **(3) Compared with the HT model** in this paper, **RG-1 and RG-2** are improved by **0.87 and 1.02**, respectively. This is because in the HT model, sentence-level and word-level transformers are introduced to encode the case–public opinion text, and as the amount of parameters increases, the model complexity increases. **(4) Compared with the GraphSum model**, the model in this paper has an improvement of **0.29 and 0.82** on **RG-1 and RG-2**, respectively, indicating that the graph structure is used to represent the relationship between documents across sentences, but the model in this paper incorporates case elements as auxiliary information. The method is more effective, can effectively reduce redundant information, and has an important guiding role in generating sentences that are closer to the topic of the document.

### 5.2. Analysis of Ablation Experiments

To verify the effectiveness of individual components such as the graph encoder module and the two-level attention module, we conduct ablation research experiments. The ablation experiments results are shown in Table 5.

**Table 5.** Ablation experiments.

Model	RG-1	RG-2	RG-L
<b>Our Model</b>	<b>32.81</b>	<b>16.78</b>	<b>27.19</b>
w/o graph encoder	29.88	13.99	20.12
w/o two-level attention	30.15	14.12	22.21

The “w/o graph encoder” represents experiments without the graph encoding module, fixing case element representation, and paragraph representation after multi-head pooling layer. The “w/o two-level attention” means experiments without two-level attention.

We directly apply token-level attention; however, the extra focus on case elements suggests that this is an easy way to combine information about case elements. Table 6 results show the effectiveness of our newly introduced module.

**Table 6.** Comparative experiments of different case element extraction methods.

Method	RG-1	RG-2	RG-L
NER	31.24	15.27	25.65
TFIDF	31.37	15.49	25.69
TextRank	32.15	16.33	26.87
<b>Our Model</b>	<b>32.81</b>	<b>16.78</b>	<b>27.19</b>

### 5.3. Comparative Experimental Analysis of Different Case Element Extraction Methods

This experiment mainly verifies the influence of obtaining case elements using different methods of abstract generation. Using TFIDF, TextRank, and named entity recognition algorithms, keywords are extracted from the case text as case elements and integrated into the model of this paper to generate abstracts. The case elements of each case in the data set constructed in this paper usually include case name, victim, suspect, case time, crime location, and other case elements, usually containing 5–8 keywords. The NER method takes the names of people and places, organization name, and time as case elements. For the TFIDF and TextRank methods, we use top-8 elements as the case elements. The results are shown in Table 6 Show.

It can be seen from the above table that (1) compared with the NER model, the RG-1 and RG-2 values of this model are improved by 1.57 and 1.51 because the NER method obtains a large amount of redundant information, which is not conducive to the learning of graph attention and results in summary performance drops. (2) When compared with the TFIDF model, the RG-1 and RG-2 values of this model are increased by 1.44 and 1.29, respectively, because TFIDF is a method based on word frequency statistics. The keywords in the text can fully express the subject information of the article, thereby improving summary performance, but words with higher word frequency in case–public opinion texts are not necessarily related to text topics. (3) When compared with the TextRank method, the model in this paper has an improvement of 0.66 and 0.45 on RG-1 and RG-2, respectively. TextRank’s method of extracting keywords has a small gap. When there are multiple documents but no case elements, it can extract keywords which serve as case elements and are integrated into the generation of auxiliary abstracts in the model of this paper.

## 6. Conclusions

In this paper, a multi-document generative summarization model based on case feature graph attention is proposed. In addition to text unit nodes, case element nodes are also introduced to construct heterogeneous graphs, which assist the model to capture complex relationships between text units. A decoder with a two-level attention mechanism is also introduced, which first pays attention to the case element nodes and then uses the attention weights to guide the attention to the text units, which can effectively deal with saliency and redundancy issues. In the next research, we will continue to explore other methods,

such as reinforcement learning-based methods, to further improve the summary quality in the multi-document summarization environment, and the model in this paper can also be applied to other tasks, such as multi-document question answering.

**Author Contributions:** Methodology, Y.H. and G.L.; Project administration, Z.Y.; Writing—original draft, Y.H. and S.H.; Writing—review/editing, Y.H. and S.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** 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 (grant numbers 2018YFC0830105, 2018YFC0830101, and 2018YFC0830100), the National Natural Science Foundation of China (grant numbers 62266027, U21B2027, and 61972186), the Yunnan provincial major science and technology special plan projects (grant number 202202AD080003), general projects of basic research in Yunnan Province (grant number 202001AT070047), and Kunming University of Science and Technology “double first-class” joint project (202201BE070001-021).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

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

## References

1. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
2. Barzilay, R.; McKeown, K.R. Sentence fusion for multidocument news summarization. *Comput. Linguist.* **2005**, *31*, 297–328. [CrossRef]
3. Filippova, K.; Strube, M. Sentence fusion via dependency graph compression. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, HI, USA, 25–27 October 2008; pp. 177–185.
4. Banerjee, S.; Mitra, P.; Sugiyama, K. Multi-document abstractive summarization using ilp based multi-sentence compression. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
5. Li, W. Abstractive multi-document summarization with semantic information extraction. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1908–1913.
6. Bing, L.; Li, P.; Liao, Y.; Lam, W.; Guo, W.; Passonneau, R.J. Abstractive multi-document summarization via phrase selection and merging. *arXiv* **2015**, arXiv:1506.01597.
7. Cohn, T.A.; Lapata, M. Sentence compression as tree transduction. *J. Artif. Intell. Res.* **2009**, *34*, 637–674. [CrossRef]
8. Wang, L.; Cardie, C. Domain-independent abstract generation for focused meeting summarization. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, 4–9 August 2013; pp. 1395–1405.
9. Pighin, D.; Cornolti, M.; Alfonseca, E.; Filippova, K. Modelling events through memory-based, open-ie patterns for abstractive summarization. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 892–901.
10. Paulus, R.; Xiong, C.; Socher, R. A deep reinforced model for abstractive summarization. *arXiv* **2017**, arXiv:1705.04304.
11. Gehrmann, S.; Deng, Y.; Rush, A.M. Bottom-up abstractive summarization. *arXiv* **2018**, arXiv:1808.10792.
12. Li, W.; Xiao, X.; Lyu, Y.; Wang, Y. Improving neural abstractive document summarization with structural regularization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 4078–4087.
13. Zhang, X.; Wei, F.; Zhou, M. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv* **2019**, arXiv:1905.06566.
14. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Mach. Learn. Res.* **2020**, *21*, 1–67.
15. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv* **2019**, arXiv:1910.13461.
16. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the International Conference on Machine Learning. PMLR; 2020; pp. 11328–11339. Available online: [https://www.google.com.hk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKewiWnjb8P\\_8AhXD6aQKHdhzB3sQFnoECAoQAQ&url=http%3A%2F%2Fproceedings.mlr.press%2Fv119%2Fzhang20ae%2Fzhang20ae.pdf&usq=AOvVaw1VKn6wia\\_Muv\\_rcuqG30sp](https://www.google.com.hk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKewiWnjb8P_8AhXD6aQKHdhzB3sQFnoECAoQAQ&url=http%3A%2F%2Fproceedings.mlr.press%2Fv119%2Fzhang20ae%2Fzhang20ae.pdf&usq=AOvVaw1VKn6wia_Muv_rcuqG30sp) (accessed on 1 January 2023).
17. Zou, Y.; Zhang, X.; Lu, W.; Wei, F.; Zhou, M. Pre-training for abstractive document summarization by reinstating source text. *arXiv* **2020**, arXiv:2004.01853.



18. Grail, Q.; Perez, J.; Gaussier, E. Globalizing BERT-based transformer architectures for long document summarization. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 19–23 April 2021; pp. 1792–1810.
19. Erkan, G.; Radev, D.R. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.* **2004**, *22*, 457–479. [[CrossRef](#)]
20. Wan, X. An exploration of document impact on graph-based multi-document summarization. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, HI, USA, 25–27 October 2008; pp. 755–762.
21. Christensen, J.; Soderland, S.; Etzioni, O. 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, Atlanta, GA, USA, 9–14 June 2013; pp. 1163–1173.
22. Tan, J.; Wan, X.; Xiao, J. Abstractive document summarization with a graph-based attentional neural model. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1171–1181.
23. Yasunaga, M.; Zhang, R.; Meelu, K.; Pareek, A.; Srinivasan, K.; Radev, D. Graph-based neural multi-document summarization. *arXiv* **2017**, arXiv:1706.06681.
24. Fan, A.; Gardent, C.; Braud, C.; Bordes, A. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. *arXiv* **2019**, arXiv:1910.08435.
25. Huang, L.; Wu, L.; Wang, L. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. *arXiv* **2020**, arXiv:2005.01159.
26. Wang, D.; Liu, P.; Zheng, Y.; Qiu, X.; Huang, X. Heterogeneous graph neural networks for extractive document summarization. *arXiv* **2020**, arXiv:2004.12393.
27. Song, Z.; King, I. Hierarchical Heterogeneous Graph Attention Network for Syntax-Aware Summarization. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2022; pp. 11340–11348.
28. Jin, H.; Wang, T.; Wan, X. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6244–6254.
29. Liu, Y.; Lapata, M. Hierarchical transformers for multi-document summarization. *arXiv* **2019**, arXiv:1905.13164.
30. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI blog* **2019**, *1*, 9.
31. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
32. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In Proceedings of the Text summarization branches out; 2004; pp. 74–81. Available online: <https://aclanthology.org/W04-1013.pdf> (accessed on 1 January 2023).
33. Liu, P.J.; Saleh, M.; Pot, E.; Goodrich, B.; Sepassi, R.; Kaiser, L.; Shazeer, N. Generating wikipedia by summarizing long sequences. *arXiv* **2018**, arXiv:1801.10198.
34. Li, W.; Xiao, X.; Liu, J.; Wu, H.; Wang, H.; Du, J. Leveraging graph to improve abstractive multi-document summarization. *arXiv* **2020**, arXiv:2005.10043.

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