Neurocomputing 460 (2021) 166–180

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Element graph-augmented abstractive summarization for legal public opinion news with graph transformer

Yuxin Huang, Zhengtao Yu *, Junjun Guo, Yan Xiang, Yantuan Xian

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China
Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, China

ARTICLE INFO

Article history:
Received 14 September 2020
Revised 24 May 2021
Accepted 4 July 2021
Available online 7 July 2021
Communicated by Zidong Wang

Keywords:
Abstractive summarization
Legal public opinion news
Element graph
Graph transformer

ABSTRACT

Automatic summarization for legal public opinion news has been an attractive research problem in recent years. Compared with the open-domain, summarization for legal public opinion news has two essential constraints: (1) the key information (e.g., the case elements) of the news should be summarized; (2) the factual errors should be avoided in the generated summary. To address these challenges, the summarizer should learn a structured representation of the news (event plan), making it better to understand the event information implied in the news. This paper proposes a novel element graph-augmented abstractive summarization model, which first constructs the structural graph by extracting elements from the source document and then produces graph representation via graph transformer network. Finally, the structural representation is taken as an essential complementary component of the conventional sequence-to-sequence model to guide the decoding process simultaneously. Furthermore, the pre-trained language model is introduced to enhance the sequential and structural encoder, which further promotes the summarization model's performance. For evaluation, we build a large-scale legal public opinion news (LPO-news) corpus. Experimental results on LPO-news and another news-oriented CNN/Daily mail dataset show that our model significantly outperforms other baselines in terms of both ROUGE scores and Bert scores. We also perform a human evaluation to demonstrate our model's effectiveness by evaluating the generated summary using several subjective metrics.

1. Introduction

Legal public opinion news usually involves legal events or cases, which spreads fast on the internet, resulting in a wildly social impact. Hence, automatically generating a concise and coherent summary for legal public opinion news is essential for rapid and effective disposal. Generally, summarization for the legal news can be formalized as a domain-specific abstractive summarization task [33,53,12], which has been wildly investigated by using sequence-to-sequence (seq2seq) frameworks [39] with attention mechanism [1]. These models fall into an encoding–decoding paradigm, where the encoder first transforms the input sequence to high-level distributed representation. The decoder generates the summary word by word under the constraints of the outputs of the encoder. However, although the summarization models based on the seq2seq framework have gained impressive performance in the open-domain, it is still challenging to generate practical summaries in the domain-specific scenario such as for legal public opinion news. As shown in Fig. 1, we present two legal-domain examples, including the source document, gold summary, and corresponding summary generated by the seq2seq model. The first example describes the case of "baby falling case in Beijing," and the seq2seq model generates a summary as *"the defendant is shopping cart"*, which expresses obvious and serious factual errors. Furthermore, for the second example, the model should summarize the main topic *"the controller has committed suicide and no longer pursued his responsibility"*, but it incorrectly focuses on the secondary aspect *"number of death"*. Therefore, in practice, to produce a practical and useful summary for the legal public opinion news, the summarization model should have some extra and critical constraints: (1) the generated summary should summarize the key information of the source document rather than the unimportant details; (2) the model should produce faithful summary rather than a fiction summary containing serious factual errors.

To alleviate these issues, many researchers seek to combine the keywords into the sequence-to-sequence model. The motivation behind these methods is that the keywords extracted from the source document can be viewed as the main aspects of the source document, which are used as additional information to guide the

* Corresponding author.
E-mail address: ztyu@hotmail.com (Z. Yu).

https://doi.org/10.1016/j.neucom.2021.07.013
0925-2312/

Source Document	国务院四川攀枝花肖家湾矿难事故调查组调查报告事发企业实际控制人于去年#月##日跳楼自杀身亡，建议不再追究其责任；市安监局监督管理三处处长等##人已被司法机关采取逮捕等措施。这起事故造成##人死亡。(The investigation team of Xiaojiawan Mine Disaster reported that the actual controller has committed suicide by jumping off a building on last year, and suggested that he should no longer be investigated for responsibility. # # people, have been arrested by judiciary authorities. The accident resulted in # deaths.)
Gold Summary	肖家湾矿难企业实际控制人自杀不再追责。(The actual controller of Xiaojiawan Mine Disaster Enterprise committed suicide and no longer pursued the responsibility.)
Generated Summary	**四川攀枝花致##人死亡矿难，##人被采取逮捕。(# # were dead in Mine Disaster in Panzhihua, Sichuan, # # were arrested)**
Source Document	北京摔婴案被告韩磊觉得在大庭广众之下和一个女人打架很丢人，看到旁边有个车就举起摔在地上，他以为是只购物车自始至终都没看车里，不知道里面有个孩子。(Han Lei, the defendant in the Beijing baby fall case, felt ashamed to fight with a woman in public. When he saw a baby car nearby, he lifted it and slammed to the ground. He thought it was just a shopping cart. He didn't look at the car and didn't know there was a child in it.)
Gold Summary	北京摔死女童案被告翻供，我不知道是婴儿车(The defendant in the case retracted his confession, and said he didn't know it was a baby carriage.)
Generated Summary	**北京摔婴案被告是购物车。(The defendant in the case is a shopping cart.)**

Fig. 1. Examples generated by conventional sequence-to-sequence model.

decoding process. Li et al. [21] proposed a key information guide network, which first extracted keywords from the source document and encoded them into high-level representation, then integrated this information into the decoder via a dual attention mechanism. Wang et al. [46] presented a topic-aware abstractive summarization model, which considered the topic words as the keywords and then included them into the convolutional sequence-to-sequence (ConvS2S) model. However, although these methods get keywords in different ways, they all treat keywords as independent tokens, ignoring the vital relationship between them. We argue that the legal public opinion news is an event-centered report and usually tells us "who (criminal) did what (crime process) to whom (victim) in when (time of the crime) and where (location of crime)", aka, case elements. Naturally, these case elements can be organized into a structural graph to reflect the event's global skeleton. As shown in Fig. 2, the source document describes the report of "Zhang Yingying's case", where her family sues the psychological social workers of Illinois University. As shown in Fig. 2, the case elements extracted from the source document are connected by different types of edges and convey explicit and concise event information to the user. Based on these observations, we argue the element graph can be viewed as a structured form of the source document and provide a global-level event plan for the user to understand the case.

This paper develops a novel graph-augmented abstractive summarization model for legal public opinion news based on the element graph. It extends the conventional seq2seq model with another separate graph encoder to produce an explicit structural representation of the source document. Specifically, the case elements, including keywords, named entities, and event triples, are first extracted from the source document. Then these elements are organized into two different graph forms: *element relational graph* (ERG) and *topic interaction graph* (TIG). As shown in Fig. 2, the nodes of ERG are connected by introducing several different types of virtual nodes (the keywords, person, location and organization, etc.), which helps capture the document-level interaction between different elements. Additionally, the topic interaction graph is constructed by estimating the similarities of different nodes, in which the representation of nodes is yielded by the topic model. The motivation is that the extracted elements can be viewed as the topics (aspects) of the source document, and the TIG can reflect the association strength between different topics, which is essential for discovering the main topic of the source document. Next, we use a dual encoder including sequence encoder and graph encoder to yield the sequential and structural representation of the source document, respectively. Finally, the decoder generates the summary word by word simultaneously constrained by sequential and structural representation. Specifically, inspired by the success of Graph Attention Network (GAT) [43], we introduce powerful Transformer architecture to fit the graph inputs (aka Graph Transformer [50]) to encoder the element graph, which can incorporate global structural information when aggregating the information from the neighbor nodes.

Furthermore, the pre-trained language model with a self-supervised objective has been proven to be effective in a wide range of natural language processing tasks, producing better representation for the input sequence. Consequently, in this paper, we introduce the pre-trained language model to initialize the element graph's nodes and enhance the embedding of sequential input. With the help of the mighty representation capacity provided by the pre-trained language model, our model further facilitates the generation of coherent, faithful, and informative summaries.

We collect a large-scale legal public opinion summarization corpus containing 123,853 news and corresponding summary for evaluation. Then, extensive experiments are conducted on this corpus, and the results show that our proposed element graph-augmented model consistently outperforms some strong baseline methods under ROUGE scores [26] and BERT scores [51]. We further carry out an automatic evaluation on another famous news summarization corpus CNN/Daily mail [13], and the results show that our model also achieves significant improvement. Moreover, human evaluation further confirms that our model can generate better summaries in terms of faithfulness, informativeness, and fluency than the models without graph encoder.

Briefly, the contributions of our work can be summarized as follows:

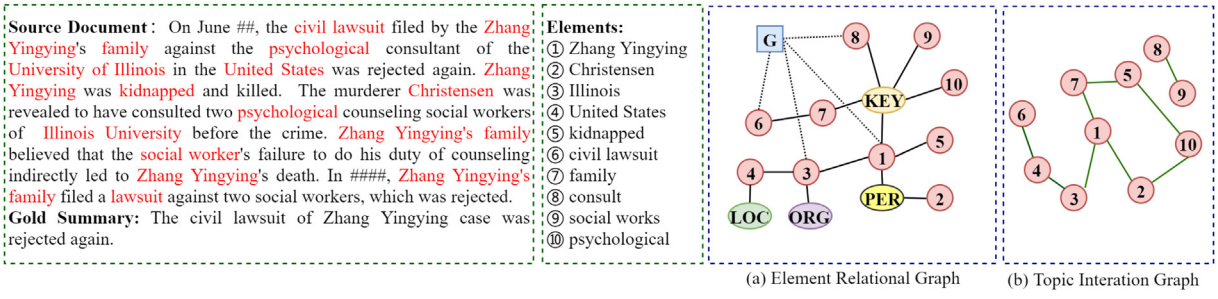

Fig. 2. An example contains the source document, gold summary, and corresponding graphical representation, in which the circles denote elements extracted from the source document. The ellipses with KEY indicate the keywords, and G is the global node connected to all element nodes to promote the flow of the information. The PER, LOC, and ORG denote the virtual nodes with person, location, and organization, respectively. (a) An element relational graph, which connects the elements by virtual nodes. (b) A topic interaction graph, which is constructed by estimating the topic similarity of the nodes.

- For the legal public opinion news summarization task, we extend the seq2seq model by introducing explicit graph representation via graph transformer network.
- We propose two kinds of graphs as global event-plan: element relational graph and topic interaction graph, which explicitly encode the elements correlation information to improve the generated summary quality.
- Furthermore, extensive experiments are conducted on legal public opinion news and CNN/Daily mail corpus. The results demonstrate that the graph-structured representation generated by the graph transformer network brings substantial improvements compared with several strong baselines.

2. Related works

2.1. Graph-to-text natural language generation

Our work falls under a larger scope of graph-to-text generation. In this direction, the previous works are mainly in the line of syntax-to-text, AMR-to-text, and knowledge graph-to-text. Bastings et al. [2] first integrated an explicit syntax graph into the neural machine translation (NMT) task by using a graph convolutional networks (GCNs) [18] to decide which aspects of syntax are beneficial for NMT. Moreover, Marcheggiani et al. [30] proposed to explore the GCN-based semantic graph to augment NMT by injecting a semantic bias into the sentence encoder and achieved better BLEU scores. Abstractive Meaning Representation (AMR) graph is another variant of semantic representation. Song et al. [38] posed an approach to leverage graph recurrent networks (GRNs) [52] as an additional encoder to obtain the meaningful representation from the AMR graph, which was useful to handle the data sparsity problem of NMT. There are also some previous studies dedicated to generating text from an AMR graph via graph encoder. Beck et al. [3] developed a Gated Graph Neural Networks(GGNNs) [25] to build a structured representation of AMR graph, in which the nodes and labels of nodes were simultaneously utilized as the GGNNs model's parameters. To capture the AMR graph structure, Damonte et al. [5] presented a structural multi-encoder mechanism by stacking the sequential encoder and graph encoder. Specifically, the graph encoder implemented by the Tree-LSTM [40] or GCNs were assembled with the bi-directional long short term memory networks (Bi-LSTMs) in different orders to capture both the sequence and structure information. Recently, generating fluent text from the knowledge graph (KG) attracted extensive attention. The KG can be drawn from a large-scale open-domain knowledge graph or solely extracted from the source document. For the former, the model read the structured graph such as resource description framework (RDF) triples of WebNLG [9] or key-value pairs of WikiBio [20], to generate the text description of the KG. Trisedya et al. [41] introduced an LSTM based encoder to directly capture the relationship between different triples for the RDF-to-text task. The other is to automatically construct the knowledge graph directly from the source document via an information extraction system (OpenIE[1]). Li et al. [24] presented an approach to generate comments based on the source document, where the long document was represented as a topic shift graph and then was encoded by GCNs network to produce a meaningful comment.

2.2. Graph-based summarization

Graph structure has long been studied in the summarization task, especially for the extractive summarization [28]. Mihalcea et al. [32] presented a graph-based model to extract salient sentences by modeling the relationship between these sentences. Wan et al. [44] further extended this approach to the multi-document summarization scenario by incorporating additional document-level relations. More recently, Wang et al. [45] developed a graph-based method to construct a heterogeneous graph including word-level and sentence-level nodes, which was then encoded by GCNs to yield the representation of the sentence. Dong et al. [7] proposed a graph-based unsupervised extractive method for the long document, which transformed the fully-connected sentence graph into a hierarchical graph by introducing section-to-section and sentence-to-section information grounded in discourse structure.

Additionally, the graph-based method also has been used for the abstractive summarization task. For the multi-document abstractive summarization (MDS) task, Yasunaga et al. [49] proposed to utilize sentence relation graph as additional structure information to indicate the salience score of different sentences, which can provide essential information for the decoder. Furthermore, Li et al. [23] developed a method to utilize the paragraph-level graph, including a similarity graph and a discourse graph to capture cross-document relations. Moreover, Li et al. [23] incorporated the graph information in the encoding and decoding process by using a special graph-informed attention mechanism. For the single document abstractive summarization task (SDS), Xu et al. [47] presented a document-level syntactic graph convolutional network. Specifically, the cross-sentence syntactic graph was first represented via GCNs to produce the structure representation, which was then integrated into the conventional sequence-to-sequence model using a gated graph attention mechanism. In addition, Fernandes et al. [8] built the graph directly from the source document. The words and entities extracted from the source document were viewed as nodes of the graph, and different relations were built between these nodes, including "co-reference", "next token" and "in sentence" etc. The motivation is that the graph can explicitly annotate important relationships between different tokens to help the decoder determine useful information. This work partly inspires our work, but our approach is quite different from theirs in the following aspects: (1) we extract the entities, keywords, and event triples from the source document as elements to construct the graph instead of utilizing all tokens; (2) we propose two different approaches to form the graph; (3) we use Graph Transformer Networks (GTNs) to encode the graph instead of GCNs, which is usually limited by the inherent over-smoothing problem.

2.3. Graph transformer

The graph neural network (GNNs) has recently attracted growing attention, which was good at modeling graph structure data. In this direction, different GNNs networks were proposed to yield high-level representation for the graph data, such as GCNs [18], GGNNs [3] etc. Recently, Graph Attention Networks(GATs) [43] was proposed to compute the node representation by attending over its neighbors via a self-attention mechanism, which allowed the model to focus on the most relevant nodes by assigning different weights to the neighbors. Inspired by GATs, Transformer [42], which has been proved to be the better architecture for sequence text encoding, was introduced to adapt the graph data. Koncel et al. [19] proposed an abstractive generation model for the scientific paper by jointly modelling the title and knowledge graph extracted from the article, in which the KG was encoded by a graph transformer network. Koncel et al. [19] further improved this model by combining both global and local representation of the graph, wherein the global node encoding mechanism allowed explicit communication between two distant nodes, and the local node encoding mechanism imposed explicit graph inductive bias. Finally, our work is partly related to [15], who presented an

[1] http://openie.allenai.org/.

abstractive summarization model with graph-augmentation and semantic-driven reward. They first extracted relevant triples as <subject, predicate, object> utilizing Stanford CoreNLP [29]. The subject and object were then treated as nodes and connected by directed edges, with predicates as the edge labels. In contrast to this approach, we extract the triples and the entities and keywords, which can indicate richer global event plan information.

3. Graph construction

This section introduces how to construct the element relational graph (ERG) and topic interaction graph (TIG). As shown in Algorithm 1, we first extract the elements, including entities, keywords, and event triples from the source document as the graph nodes. Then, we link these nodes by introducing several types of virtual nodes to form the ERG or estimate the similarity between different nodes to create the TIG. We will introduce the detailed process below.

Algorithm 1. Graph construction

Input: The document $\mathbf{x}$, the topic number T, and the threshold γ.
Output: Element Relational Graph G^e and Topic Interaction Graph G^t

```
Segment document x into sentences S.
Perform keywords extraction, named entity recognition operation, get keyword set K and entities set E.
for each sentence s in S do
  if s contains entities e then
    Extract the event triples
if ERG G^e then
  Connect k to v_KEY, e to v_PER, v_LOC or v_ORG, and build connection between the event triples.
  for each sentence s in S do
    if s contains (e_i, e_j) or (e_i, k_j) then
      Connect (e_i, e_j) or (e_i, k_j);
  Connect k, e and event triples to v_G.
  return G^e
else if TIG G^t then
  Use topic model to yield the node's representation;
  for nodes v_i and v_j do
    Calculate similarity: sim_ij = cos_sim(v_i, v_j)
    if sim_ij > γ then
      Connect v_i and v_j
  return G^t
```

Given a document $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, where N is the number of words. We first perform named entities recognition and keywords extraction from the source document with the off-the-shelf tools such as LTP[2] (for Chinese) or Stanford CoreNLP[3] (for English). Then we further parse the document with a dependency syntactic parsing tool to get the event triples, where the subject, object, and predicate are also treated as the nodes of the graph. The motivation is that we argue the named entity and keywords only reflect the static view of the event, and the event triples can provide the dynamic relationship between these elements, which is an effective supplement to assist the model in understanding the event. As a result, we get an element collection including keywords, named entities, and event triples. Then we further introduce how to build the edges between these elements.

[2] https://github.com/HIT-SCIR/ltp.
[3] https://github.com/stanfordnlp/CoreNLP.

We construct the ERG based on the following observations: the same elements may appear in different sets and can be directly connected by themself. For example, in Fig. 2, *Zhang Yingying* is not only a keyword, but also a named entity, and a subject in the event triples (*Zhang Yingying*, *Kidnapped*). Hence, The elements sequence (v_1, v_{PER}, v_2, v_5) in Fig. 2(a) denotes that *Zhang Yingying is a person and be kidnapped by another person Christensen*. To this end, we take four steps to construct the edges by introducing several virtual nodes (v_{KEY}, v_{PER}, v_{LOC}, v_{ORG}, and v_G). First, the keywords, entities are directly connected to the corresponding virtual nodes. Then the subject in event triples is linked to the predicate and object. Additionally, we establish the connection between different entities or the entities and keywords, if they co-occurred in the same sentence. Finally, a global virtual node v_G linking to all non-virtual elements is introduced to promote the information flow between different elements. The final result of these operations is a connected, unlabelled graph $G = (V, E)$, where V is a list of elements and the nodes, E is an adjacency matrix describing the directed edges. With this, we can capture the interaction between event-related elements at the document level.

We also build the *topic interaction graph* (TIG) to reveal the topic relation of the extracted elements. The motivation behind this is that the source document can be decomposed into several topic-centered clusters, each of which can be represented by the extracted elements. Based on these observations, we first utilize a topic model to yield the topic distribution, wherein the topic model can be implemented by statistics based LDA model [4] or deep learning based neural topic model [31,6]. These model can generate document topic distribution $\boldsymbol{t}_x \in \mathbb{R}^T$ at document-level and the words topic distribution $\mathbf{t}' = (t'_1, \ldots, t'_N)$ at word level, where T denotes the number of the topic and $t'_i \in \mathbb{R}^T$ represents the topic distribution of i-th word in document $\mathbf{x}$. Considering that the word x_i may appear in multiple documents, to generate the unique representation of word x_i in document $\mathbf{x}$, we adopt the same strategy as Narayan [34], taking $t'_i \otimes t_x$ as the topic representation of x_i. The motivation for this operation is that the t_x denotes the global topic information of the document $\mathbf{x}$, and the t'_i captures local topic information in itself. Thus $t'_i \otimes t_x$ can jointly utilize the global and local topic information to generate better representation. Finally, we link the elements if the cosine similarities of arbitrary two nodes exceed the threshold γ, which is an essential factor for the topic interaction graph to determine the graph's sparsity. Thus we will give a detailed analysis in Section 6.2.6.

4. Model description

This section gives a detailed description of the proposed element-graph augmented abstractive summarization framework, as shown in Fig. 3. Concretely, it is composed of two transformer-based encoders including a vanilla sequential transformer and a graph transformer, which independently consumes the source document $\mathbf{x} = \{x_i\}$, and corresponding graph $G = \{v_i\}$ to yield the sequential context and structural context. The sequential transformer decoder is then adopted to produce the summary word by word constrained by both the sequential and structural context. In the following, we will introduce them in turn.

4.1. Sequential transformer encoder

4.1.1. Vanilla transformer

Formally, the sequential encoder reads the input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ and yields the context representation $\mathbf{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N)$. Note that, the encoder can be realized by recurrence neural networks (RNNs) [39], convolution neural networks (CNNs) [10] or self-attention [42]. In this paper, we

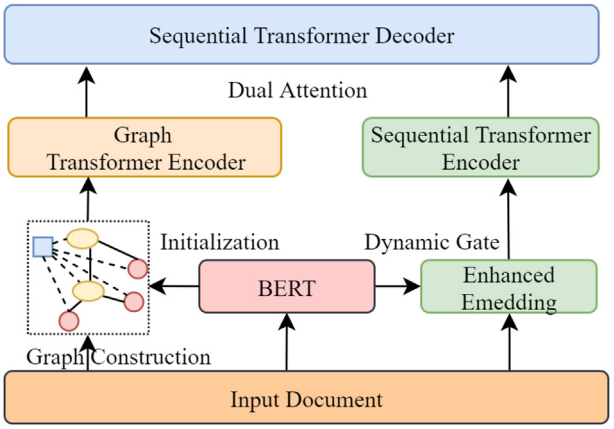

Fig. 3. Our proposed model is divided into three components: (1) sequential transformer encoder to yield the sequential context; (2) graph transformer encoder to generate the structural representation of the source document by reading graph input; (3) sequential transformer decoder with a dual attention mechanism. Specifically, the pre-trained language model such as BERT is applied to boost the performance in two ways: enhancing the sequential encoder's embedding and initializing the nodes of the element graph.

implement our model using self-attention based Transformer architecture, which have achieved state-of-the-art performance in many text classification and text generation tasks. Typically, for input sequence $\mathbf{x}$, each token x_i is mapped into a continuous space by looking up the embedding matrix as follows:

$$\mathbf{e}_i = \text{emb}(x_i) \tag{1}$$

where emb is the mapping function.

Then, a multi-head attention module is applied to produce the l-th layer high-dimensional representation $\hat{\boldsymbol{h}}_i^l$ by attending over the $l-1$ layer representations of $\mathbf{x}$:

$$\hat{\boldsymbol{h}}_i^l = \sum_{j\in\mathbf{x}} \alpha_{i,j} \mathbf{W}_e \boldsymbol{h}_j^{l-1} \tag{2}$$

here $\mathbf{W}_e$ is the training parameters of projection matrices and $\boldsymbol{h}_j^0 = \boldsymbol{e}_j$. Specifically, $\alpha_{i,j}$ is the normalized attention weights between $\boldsymbol{h}_i^{l-1}$ and $\boldsymbol{h}_j^{l-1}$ which can be formed as a scaled dot-product operation as following:

$$\begin{aligned} \alpha_{i,j} &= \text{softmax}(u_{i,j}) = \frac{\exp(u_{i,j})}{\sum_{j\in\mathbf{x}} \exp(u_{i,j})} \\ u_{i,j} &= \frac{(\mathbf{W}_K \boldsymbol{h}_j^{l-1})^\top \mathbf{W}_Q \boldsymbol{h}_i^{l-1}}{\sqrt{d_k}} \end{aligned} \tag{3}$$

where $\mathbf{W}_K$ and $\mathbf{W}_Q$ are trained parameters. Note that, the dot-product is scaled by the scaling factor $\frac{1}{\sqrt{d_k}}$, where d_k is the dimension of $\boldsymbol{h}_i$.

Then, the multi-head attention module is composed of two additional sub-layers to form a standard transformer block as following:

$$\begin{aligned} \boldsymbol{h}_i^l &= \text{LayerNorm}\left(\tilde{\boldsymbol{h}}_i^l + \text{LayerNorm}\left(\hat{\boldsymbol{h}}_i^l\right)\right) \\ \tilde{\boldsymbol{h}}_i^l &= \text{FFN}\left(\text{LayerNorm}\left(\hat{\boldsymbol{h}}_i^l\right)\right) \end{aligned} \tag{4}$$

where LayerNorm is a layer normalization function; and FFN is a feed-forward network with ReLU as activation function. Finally, the $\boldsymbol{h}_i^l$ is considered as the output of the encoder at l-th layer.

4.1.2. Enhanced embedding

Recently, different pre-trained language models are proposed to learn the implicit linguistic rules and common sense knowledge from massive unlabeled data via a self-supervised learning objective. We argue that using the knowledge implied in the pre-trained language model as context embedding is useful for text representation [54]. Motivated by this observation, we present an enhanced embedding mechanism by utilizing the pre-trained language model(For simplicity, we will refer to it as BERT in the rest of this paper) to boost sequential embedding by introducing a dynamic gate mechanism into the vanilla transformer encoder. More specifically, the input sequence is first encoded by the BERT model as:

$$\mathbf{H} = \text{BERT}(\mathbf{x}) \tag{5}$$

where $\mathbf{H}$ is the output of the last layer of BERT and $\boldsymbol{h}_{CLS}$ is the representation of the classification token CLS, and it is then used as the representation of input sequence $\mathbf{x}$. However, although BERT can provide useful knowledge for the downstream tasks, fine-tuning an entire new pre-trained model is also challenging due to the large-scale parameters (more than 110 million for BERT_Base model) and catastrophic forgetting problem. Hence, in this paper, we adopt a simple and effective adapter-based fine-tuning mechanism [14], in which the parameters of BERT are fixed, and an adapter is added on the top of BERT to transfer the BERT output to a new representation to adapt the downstream task. Specifically, the output of BERT $\boldsymbol{h}_{CLS}$ is transferred by introducing a linear network as the adapter:

$$\boldsymbol{z} = \mathbf{W}_z \boldsymbol{h}_{CLS} + b_z \tag{6}$$

where $\boldsymbol{z}$ denotes the transferred representation of input sequence $\mathbf{x}$, $\mathbf{W}_z \in \mathbb{R}^{d\times l}$ and $b_z \in \mathbb{R}^{d\times 1}$ are learning parameters with d (512 in this paper) and l (768 in this paper) being the embedding dimension and BERT output dimension, respectively.

Inspired by [53], we introduce a dynamic gate mechanism to combine the BERT representation $\boldsymbol{z}$ and the word embedding $\boldsymbol{e}_i$. The starting for this strategy is an observation that different words have different needs of the BERT representation. Concretely, a dynamic gate λ_i is first calculated based on the transferred BERT representation $\boldsymbol{z}$ and i-th token's embedding $\boldsymbol{e}_i$:

$$\lambda_i = \sigma\left(\mathbf{W}_{\mathbf{g1}} \boldsymbol{e}_i + \mathbf{W}_{\mathbf{g2}} \boldsymbol{z}\right) \tag{7}$$

in which, σ is a logistic sigmoid function, and $\mathbf{W}_{\mathbf{g1}} \in \mathbb{R}^{1\times d}$ and $\mathbf{W}_{\mathbf{g2}} \in \mathbb{R}^{1\times d}$ are learning parameters.

Then, the enhanced embedding is yielded from a linear combination of $\boldsymbol{e}_i$ and BERT output $\boldsymbol{z}$ as:

$$\tilde{\boldsymbol{e}}_i = \boldsymbol{\lambda}_i \otimes \boldsymbol{e}_i + (\mathbf{1} - \boldsymbol{\lambda}_i) \otimes \boldsymbol{z} \tag{8}$$

where $\otimes$ represents an element-wise multiplication.

4.2. Graph transformer encoder

The element graph $G = (V, E)$ described in Section 3 can provide explicit global structural information for the decoder. Inspired by [15,19], we encode the graph to produce the hidden representation of node v_i by aggregating its neighbors following self-attention strategy. Specifically, we extend the Transformer framework to the graph structure data by considering different nodes' relations. Similar to the vanilla transformer, we compute the representation of node v_i by a multi-head attention operation:

$$\begin{aligned} \hat{\boldsymbol{m}}_i^l &= \boldsymbol{m}_i^{l-1} + \sum_{j\in\mathcal{N}_i} \alpha'_{i,j} \mathbf{W}_v \boldsymbol{m}_j^{l-1} \\ \alpha'_{i,j} &= \text{softmax}\left(u'_{i,j}\right) = \frac{\exp\left(u'_{i,j}\right)}{\sum_{j\in\mathcal{N}_i} \exp\left(u'_{i,j}\right)} \\ u'_{i,j} &= \frac{\left(\mathbf{W}'_K \boldsymbol{m}_j^{l-1}\right)^\top \mathbf{W}'_Q \boldsymbol{m}_i^{l-1}}{\sqrt{d_m}} \end{aligned} \tag{9}$$

where $\mathcal{N}_i$ denotes the directly neighbours of node v_i, $\boldsymbol{m}_i^{l-1}$ denotes the representation of node v_i at $l-1$ layer and d_m is the dimension of $\boldsymbol{m}$. To get the directly neighbours of v_i, we construct a binary matrix $\mathbf{R} \in \mathbb{R}^{N\times N}$ based on the ERG or TIG graph described in Section 3, in which the $r_{i,j}=1$ indicates a connection between v_i and v_j, and vice versa. The matrix is then used as the mask matrix of self-attention to control which nodes are visible or need to be masked in the process of self-attention calculation. Compared with vanilla transformer, the key difference of the graph transformer is that only the directly connected nodes of v_i (including itself) are aggregated to current node instead of all nodes in the graph. These operations can implicitly encode the structural information into the node representation. Note that, the additional transformer sub-layers are also introduced to form the transformer block, as shown in Eq. (4).

4.2.1. Node initialization

As stated above, the nodes of the element graph contain named entities, keywords, and event triples (also virtual nodes in ERG). We design a node embedding function as $\boldsymbol{e}_i^v = f(v_i)$ to initialize the nodes and produce d-dimensional embedding. Usually, $f(\cdot)$ can be realized by different strategy such as *random initialization*, *pre-trained word embeddings*, *RNNs* or *BERT*. In this paper, we take each node as an independent input sequence and send it to the BERT model:

$$\mathbf{H}_i^v = \text{BERT}(v_i) \tag{10}$$

where the CLS token's representation of last layer is selected as the embedding of the node v_i.For simplicity, we abbreviate it as $\boldsymbol{e}_i^v$. Note that, the $\boldsymbol{m}_i^0 = \boldsymbol{e}_i^v$. Like Section 4.1.2, we just use BERT model as the initialization component of the graph nodes, and its parameters are fixed in training process. The detailed analysis of different choices of $f(\cdot)$ will be given in Section 6.2.2.

4.3. Sequential transformer decoder

The decoder in this paper follows the same architecture of Transformer decoder. At each decoding time step t, the summary token y_t is generated jointly constrained by the source document and element graph utilizing a dual attention mechanism.

4.3.1. Attending to the sequential

At decoding step t, the sequential context $\boldsymbol{c}_t^s$ is produced by attending to the outputs of the sequential transformer:

$$\begin{aligned} e_{t,i}^s &= \mathbf{V}_a^\top \tanh\left(\mathbf{W}_a[\boldsymbol{h}_i; \boldsymbol{s}_{t-1}]\right) \\ \alpha_{t,i}^s &= \frac{\exp\left(e_{t,i}^s\right)}{\sum_{i=1}^N \exp\left(e_{t,i}^s\right)} \\ \boldsymbol{c}_t^s &= \sum_i \alpha_{t,i}^s \boldsymbol{h}_i \end{aligned} \tag{11}$$

where $\boldsymbol{s}_{t-1}$ denotes the decoder hidden state at $t-1$ time step, and tanh is activation function. $\mathbf{V}_a^\top, \mathbf{W}_a$ are also trainable parameters.

4.3.2. Attending to the graph

We also compute the graph context $\boldsymbol{c}_t^v$ in a similar way:

$$\begin{aligned} e_{t,j}^v &= \mathbf{V}_b^\top \tanh\left(\mathbf{W}_b\left[\boldsymbol{m}_j; \boldsymbol{s}_{t-1}\right]\right) \\ \alpha_{t,j}^v &= \frac{\exp\left(e_{t,j}^v\right)}{\sum_{j=1}^J \exp\left(e_{t,j}^v\right)} \\ \boldsymbol{c}_t^v &= \sum_j \alpha_{t,j}^v \boldsymbol{m}_j \end{aligned} \tag{12}$$

where J is the length of the element graph.

4.3.3. Token prediction

Lastly, at decoding step t, the graph context $\boldsymbol{c}_t^v$ and sequential context $\boldsymbol{c}_t^s$ are merged with decoding hidden state $\boldsymbol{s}_t$, and then mapped to an output distribution over the target vocabulary:

$$p(y_t|y_1,\ldots,y_{t-1}) = \text{softmax}\left(\mathbf{W}_p \tanh\left(\mathbf{W}_t\left[\boldsymbol{s}_t; \boldsymbol{c}_t^v; \boldsymbol{c}_t^s\right]\right)\right) \tag{13}$$

where $\mathbf{W}_p$ and $\mathbf{W}_t$ are trainable parameters. [;] is concatenation operation. Note that, different from the previous works in this direction [19,15], we do not utilize the copy mechanism [36] to copy words both from the graph and sequential input. We find many repeated words are generated in summary, which seriously affects the readability and fluency of the generated summary.

We optimize our model with the negative log-likelihood loss function between the generated summary $\mathbf{y}$ and the ground-truth $\hat{\mathbf{y}}$:

$$\mathscr{L} = -\sum \log p_\theta(\boldsymbol{y}|\mathbf{x}, G) \tag{14}$$

where the loss function is equivalent to maximizing the conditional probability of summary $\boldsymbol{y}$ given parameters θ, source document $\mathbf{x}$, and the corresponding graph G.

5. Experimental setup

5.1. Datasets

We verify our model on a large legal public opinion news corpus and another a popular document-level news summarization dataset CNN/Daily mail corpus (CNN/DM) [13]. We will introduce them in turn.

We construct the legal public opinion news corpus (LPO-news) by collecting article-summary pairs from Sina Weibo, the largest micro-blog website in China. In detail, we manually filter the collected samples and annotate the data related to the Legal domain. We also exclude the samples with article length less than ten and summary length less than five and replace the number in the sample with # to reduce the complexity. Eventually, these operations result in a dataset with a total of 123,853 samples, which are further randomly split into a training (121,853), a development (1,000), and a test set (1,000). For the CNN/DM corpus, we use the same pre-process script provide by [33,36] to yield the same non-anonymous version corpus,[4] which contains 287,227 training samples, 13,368 validation samples, and 11,490 test samples, as shown in Table 1.

5.2. Evaluation metrics

Following previous works, we evaluate our model with ROUGE scores [26], which determine the summary quality by counting the overlapping units between gold summary and generated summary, as shown in formula (15):

$$\text{ROUGE-N} = \frac{\sum_{S\in\{\text{Gold}\}}\sum_{n-gram\in S} Count_{match}(n-gram)}{\sum_{S\in\{\text{Gold}\}}\sum_{n-gram\in S} Count(n-gram)} \tag{15}$$

where $n-gram$ denotes different granularity units, such as unigram, bigram and trigram. {Gold} represents the standard summary set (gold summary). $Count(n-gram)$ is the $n-gram$ number in the gold summary and the $Count_{match}(n-gram)$ denotes the number of $n-gram$ co-occurring in the generated summary and the gold summary. In this paper, we use the pyrouge script[5] to provide the F1-Score of ROUGE-1, ROUGE-2, and ROUGE-L, which measures the word overlapping, bigram overlapping, and longest common sequence, respectively. Specifically, in the rest of the paper, they will

[4] https://github.com/bisee/cnn-dailymail.

Table 1
Data statistics for LPO-news (Top),and CNN/DM (Bottom). #(examples) denotes the number of samples. AvgArticleLen and AvgSummLen represent the average length of the article and summary, respectively.

Dataset	Training	Development	Test
#(examples)	121,853	1,000	1,000
AvgArticleLen	70.40	70.14	71.20
AvgSummLen	12.08	12.11	12.31
#(examples)	287,227	13,368	11,490
AvgArticleLen	751	769	778
AvgSummLen	55	61	58

be abbreviated as RG-1, RG-2, and RG-L.

Recently, Zhang et al. [51] proposes a new evaluation metric for the text generation task, aka, BERTScore. It evaluates the quality of the text generation model by calculating the semantic-based similarity between the generated sentence and gold summary, and the semantic representations are first produced by the pre-trained BERT model.

$$\begin{aligned} R_{\text{BERT}} &= \frac{1}{|x|}\sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \\ P_{\text{BERT}} &= \frac{1}{|\hat{x}|}\sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \\ F_{\text{BERT}} &= 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \end{aligned} \tag{16}$$

in which x and $\hat{x}$ represent the gold and generated summary, respectively. $\mathbf{x}_i$ and $\hat{\mathbf{x}}_j$ denote the contextual representation of i-th token in x and j-th token in $\hat{x}$, which are generated by pre-trained BERT model. We take R_{BERT} as an example, each token in x is first matched to the most similar token in sentence $\hat{x}$ to produce the maximize matching score of x_i, then R_{BERT} is calculated by averaging the maximize matching score.

In summary, compared with the popular n-gram-based evaluation metrics, the BERTScore directly evaluates the semantic equivalence instead of surface-form similarity based on word matching. Hence, we further assess the generated summary's semantic quality by using F_{BERT}.[6]

5.3. Implementation details

We build our model based on Fairseq framework [35]. In detail, we train our model on NVIDIA P100GPU for 50 epochs with a batch size of 4096 tokens. The optimizer is Adam [16] with learning rate $7e-4, \beta_1 = 0.9$ and $\beta_2 = 0.98$. The Warm-up strategy is applied over the first 4000 steps, and also the learning rate decay strategy as mentioned in [42] is used, and the minimum learning rate is set to $1e-9$. The label smoothing regularization strategy is applied with smoothing factor 0.1. To make use of a larger batch, the gradient accumulation strategy is adopted with 4 steps. Not that, to alleviate the out of vocabulary(OOV) problem, we pre-process the input with BPE script[7] [37], and produce a vocabulary with 10,920 tokens. We share the embedding layer between the encoder and decoder to reduce the parameters. 6-layers Transformer blocks are used for both encoder and decoder with 8 heads, 2048 for the feed-forward layer. To alleviate the over-fitting problem for the graph transformer, we reduce the number of heads and the feed-forward dimensions to 4 and 1024, respectively. During inference, the standard beam search strategy is used with beam size 5, and the decoding process is stopped whenever $< eos >$ token is predicted, or the output exceeds the maximum length of 30 for LPO-news. For CNN/DM, the minimum decoding length is set to 30 and the maximum length is set to 100. For the pre-trained model, we utilize the BERT-Base model, which is implemented based on transformer with 12-layers, 768 hidden states, 12-heads, and 110 M parameters[8] for both LPO-news and CNN/DM.

5.4. Comparison methods

For LPO-news, we introduce several sentence-level abstractive summary models, which have achieved impressive performance in recent years: **SEASS** [53] introduces a selective encoding mechanism to distill the salient information of source document; **CGU** [27] adds a global encoding module into the encoder-decoder framework to improve the representation of the source input. Note that we re-implement these two models based on transformer architecture with BERT enhanced embedding module, denote as SEASS-T and CGU-T, respectively. Furthermore, we present two keywords-guided baseline models. **KIGN** [48] extracts key information from the source document and integrates them into the decoding process by utilizing a multi-view attention mechanism; **KCS** [22] further improves the KIGN by introducing a co-selective module to incorporate the information of the source document and keywords.

For CNN/DM, we include the following abstractive models for comparison: **PG+COV** model [36] includes a pointer-generator mechanism and a coverage mechanism to alleviate the OOV and word repetition problem; **Bottom-Up** [11] introduces a content selector into the transformer-based encoder-decoder architecture to assist with the model to capture the main aspect of the source document, which achieves the state-of-the-art performance. In addition, we also present two graph-based summarization models. **StruSumm** [8] treats each word in the source document as node of the graph and connects them by adding several special edges; **ASGARD** is produced by Huang et al. [15], which constructs two variants graph to enhance the decoder, in which the ASGARD-DOC build the graph at document-level by directly connecting the extracted triples using the predicates as edge label, and the ASGARD-SEG constructs the graph at the paragraph-level to capture the interaction across paragraphs.

We also describe the models presented in this paper as: **TF** is the vanilla encoder-decoder model based on transformer; **TF+BERT** denotes integrating the Bert enhanced embedding module into the transformer; **ERG-GAT** and **ERG-GTN** denote the element graph is encoded by graph attention network (GATs) [43] or Graph Transformer [50] (GTNs), respectively. Meanwhile, **TIG-GAT** and **TIG-GTN** indicate using the topic interaction graph.

6. Results and analysis

6.1. Results

6.1.1. Results on LPO-news corpus

Table 2 presents our proposed models' results and several strong baselines on the LPO-news corpus. We first notice that our ERG-GTN model obtains scores of 44.28 ROUGE-1, 36.64 ROUGE-2, and 43.15 ROUGE-L and substantially outperforms various previous methods. The ERG-GTN model especially achieves significant improvements (4.34 ROUGE-1, 7.15 ROUGE-2, and 5.53 ROUGE-L) compared with the TF-BERT model, suggesting the effectiveness of taking the element graph as the global information to augment the seq2seq model. Note that the TF-BERT model yields significantly higher ROUGE scores than the TF model, demonstrating the benefits of introducing the pre-trained model to enhance the seq2seq model. Moreover, we notice that in contrast to the TF model, the PG+COV model achieves better performance

[6] https://github.com/Tiiiger/bert_score.
[7] https://github.com/glample/fastBPE.
[8] https://github.com/google-research/bert#pre-trained-models.

Table 2
The results of different models on LPO-news test set.

Models	RG-1			RG-2			RG-L			BERTScore
	P	R	F1	P	R	F1	P	R	F1	–
SEASS-T	43.68	39.70	41.63	30.22	26.84	28.58	39.48	36.32	37.68	74.32
CGU-T	42.91	38.88	40.95	31.39	27.34	29.12	38.83	35.29	36.90	73.48
KIGN	45.38	40.11	42.51	33.17	30.25	31.71	41.61	37.53	39.49	75.71
KCS	44.28	38.95	41.95	33.66	28.48	30.88	42.29	39.49	40.90	75.03
PG+COV	40.01	37.03	38.26	30.61	26.44	28.33	37.88	33.90	35.82	74.46
Bottom-Up	40.36	37.52	38.71	31.09	26.75	28.69	37.94	34.34	36.03	74.27
TF	38.56	34.71	36.42	29.36	25.39	27.13	36.48	32.58	34.39	74.04
TF-BERT	42.39	37.82	39.94	32.27	27.43	29.49	39.39	35.87	37.62	74.66
ERG-GAT	45.06	42.22	43.36	35.49	33.55	34.24	43.55	39.69	41.53	75.36
ERG-GTN	45.88	43.17	**44.28**	38.02	35.81	**36.64**	45.60	40.93	**43.15**	78.29
TIG-GAT	45.47	42.05	42.80	35.67	33.51	34.47	44.34	40.01	42.03	76.55
TIG-GTN	45.63	42.19	43.96	37.59	35.37	36.24	44.21	39.37	41.50	**78.98**

attributed to the pointer mechanism, and the Bottom-Up model, which is an improved PG+COV model by introducing a content selector module, achieve a slight boost than the PG+COV model. Moreover, we find that the PG+COV and Bottom-Up models produce poor performance compared with other baseline models. We attribute the drop to two reasons: one is that the LSTM model is used instead of Transformer in PG+COV and Bottom-Up. In addition, the point mechanism and bottom-up attention mechanism are more suitable for the document-level input. Furthermore, we find that the keywords-guided models such as KIGN and KCS achieve significant increment over the TF model, even the TF-BERT model, which indicates that the keywords are effective for the decoder to capture the main aspects of the source document and generate a better summary. However, there is still a big gap between KIGN, the best keywords-based model, and our proposed ERG-GTN model. Specifically, the ERG-GTN model achieves 1.77, 4.93, and 3.66 improvements over the KIGN model in terms of ROUGE-1, ROUGE-2, and ROUGE-L, respectively. We hold that it is more useful to organize the keywords into a graph and encode it with graph encoders, representing the source document's structural information. Moreover, we observe that the ERG model performs better than the TIG model. We hypothesize the possible reason is that the ERG model constructs the graph directly and further introduces the global node to promote information flow between the unconnected nodes. In contrast, the TIG model reflects the interaction between different topics by introducing topic similarity, which may be influenced by the error accumulation problem raised by the topic model. Additionally, the GTNs model achieves better performance than GATs, regardless of using ERG or TIG. We hypothesize that the superior performance of GTNs stems from the modules of Layer Normalization and residual connection, which are essential to avoid the problems of over-fitting and vanishing gradient caused by the deep transformer model.

Furthermore, we report the BERTScore results to estimate the quality of the generated summaries. As presented in Table 2, all variant models gain significant improvement compared with the TF and TF-BERT model that do not use graphs in terms of the BERTScore metric, implying that leveraging element graph improves the quality of the generated summary. We assume the possible reason is that the element graph empowers the decoder to grasp the global event information, which promotes the decoder to yield semantic relevant summaries. Among the model variants, the ERG-GTN and TIG-GTN achieve better performance than GATs based models, further indicating the graph transformer network's superiority. Therefore, it is safe to conclude that our model's performance can be significantly improved by introducing the element graph into the sequence-to-sequence framework both on the ROUGE score and BERT score metric.

6.1.2. Results on CNN/daily mail corpus

We further evaluate our model on the CNN/DM corpus. As displayed in Table 3, we observe similar trends compared with the LPO-news corpus, which indicates that our proposed element graph can be well extended to another event-centered news dataset. First, We see that the TF-BERT achieves similar performance compared with the TF model. We argue the possible reason is that due to the samples in CNN/DM is long (more than 750 words), the outputs of BERT may be noisy. In addition, CNN/DM corpus is larger (more than 280 thousand), the model can learn good word embedding solely depend on the corpus itself. Moreover, the SEASS-T and CGU-T models, which distill the salient information of the input at the encoder by using the document's representation, do not achieve better performance on CNN/DM corpus compared with the TF model. We conjecture this happens because that the samples in CNN/DM are long (average 766 words), and the document-level representations may be noisy, so using an inaccurate representation as the criteria for filtering information may harm the model's performance on CNN/DM. In addition, the keywords-based models KIGN and KCS achieve a slight improvement compared with the TF model. We attribute the improvement to the fact that the keywords extracted from the source document can indeed provide key information to the encoder. However, it also can be seen that the methods based on the element graph (ERG-GTN and TIG-GTN) can further model the relationship between the keywords, thereby achieving better performance. Moreover, compared with the TF-BERT model, our proposed model also achieves consistent improvement by introducing the element graph, which further demonstrates the effectiveness of our model. Moreover, we notice that our model yields higher ROUGE scores than the ASGARD model, which extracts the triples as the nodes and constructs the graph differently. This signifies that our model extracts entities, keywords, and event triples as the graph's nodes, expressing more abundant event information than using only the triples. We also find that the models based on the ERG perform better than the TIG models, which is similar to the LPO-news results, further indicating that ERG better captures the global event plan than TIG.

6.2. Analysis

6.2.1. Ablation analysis

We conduct an ablation study on the LPO-news corpus to investigate individual modules' effectiveness in our best model ERG-GTN. We test the modules in two ways: we remove the graph encoder from the ERG-GTN model, which means that the model degenerates to TF-BERT; we remove the sequential encoder, the decoder generates the summary solely relying on the graph encoder. Table 4 summarizes the results. We see that the ERG-GTN

Table 3
The results of different models on CNN/DM test set.

Models	RG-1	RG-2	RG-L
PG+COV	39.53	17.28	36.28
SEASS-T	38.37	16.42	35.35
CGU-T	38.11	16.29	35.13
Bottom-Up	41.22	18.68	38.34
KIGN	39.32	17.89	37.16
KCS	39.33	17.47	36.74
StruSumm	38.10	16.10	33.20
ASGARD-DOC	40.38	18.40	37.51
ASGARD-SEG	40.09	18.30	37.30
TF	39.10	17.24	36.65
TF-BERT	40.32	17.63	36.67
ERG-GAT	40.76	17.81	37.40
ERG-GTN	**42.45**	**19.62**	38.18
TIG-GAT	40.09	17.92	37.63
TIG-GTN	41.53	18.89	**38.78**

Table 4
Ablation analysis on the LPO-news test set.

Models	RG-1	RG-2	RG-L
ERG-GTN	44.28	36.64	43.15
w/o graph encoder	39.94	29.49	37.62
w/o sequential encoder	28.39	18.04	26.77

model shows a serious decline of 9.8%, 19.5%, and 12.82% after removing the graph encoder, demonstrating that the graph encoder is an effective supplement to the sequential encoder. Furthermore, we notice that the sequential encoder is more important than the graph encoder, leading to more severe degradation after the sequential encoder is removed. We argue that the element graph only contains the backbone information of the source document, while the sequential encoder expresses the detailed information, which is essential for understanding the source document.

6.2.2. The impact of the node initialization methods

We further investigate the influence of different graph node initialization methods based on our best model ERG-GTN. In detail, we design four node initialization approaches: (1) **Random**: the nodes are randomly embedded by an embedding matrix, which are learned along with the model. Note that each node is embedded to a 512-dimensional vector by introducing an individual node-level vocabulary; (2) **Glove**: the nodes are initialized by the pre-trained 300 dimensional glove vector. In particular, if the node is composed by multiple words, the node embedding is the average of all words; (3) **LSTM** denotes that we introduce a 1-layer BI-LSTM model to encode the nodes, where each node is treated as a sequential input of the BI-LSTM model and shares the embedding layer with the sequential encoder; (4) **BERT** represents the nodes are initialized with the outputs of the BERT model, as described in Section 4.2.1. Then we train the model with different graph node initialization strategies mentioned above on LPO-news corpus, and present the results of the test set in Fig. 4.

We notice that the BERT model performs best among all the variant models. The BERT model gains an improvement of 4.21 points on ROUGE-1, 5.27 points on ROUGE-2, and 2.45 points on ROUGE-L compared with the Random method. It indicates that the BERT model can map the graph nodes to better representations in high-dimensional space by introducing pre-trained knowledge. The Glove model also utilizes pre-trained knowledge but produces the same representation for the same word, regardless of its context information. However, the BERT model generates contextual representations for different words, which is more useful for the

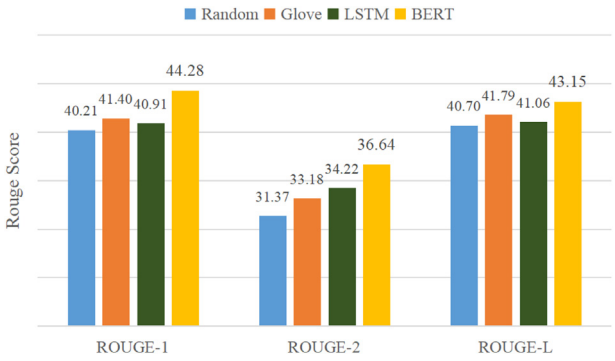

Fig. 4. Results of ERG-GTN model with different graph node initialization methods on the LPO-news test set.

model, leading to significant ROUGE score improvements compared with the Glove model.

6.2.3. The impact of the number of graph nodes

In this section, we perform an inspection on the impact of the number of graph nodes, measuring by using the ROUGE-2 score for the LPO-news test set. As shown in Fig. 5, we divide the test set into four groups and present the ERG-GTN and TIG-GTN model results. As seen, all model achieves best scores when the number of nodes is between 10 and 15. The possible reason is that the average input document length in LPO-news is about 70, and 10–15 nodes can completely cover the source document's main idea. We note that the model gains a slight drop when there are few nodes in the graph. We argue that if a document has few nodes, which also indicates its structure is simple and can be easily understood for the model. We further notice that the ERG-based model is more susceptible to the graph's size than the TIG-based model. We hypothesize that the nodes in the ERG are directly connected by the virtual nodes, thus are sensitive to the noise information causing by the information extraction system.

6.2.4. The impact of topic model

In this section, we investigate the impact of different topic models on the summarization task's performance. Specifically, we introduce three popular topic models:

- LDA [4] is a traditional topic model based on Gibbs Sampling, which can produce the document topic distribution and word topic distribution.
- NVDM [31] is a neural topic model based on variational auto-encoder (VAE) [17]. The model maps the bag of words (BOW) representation of document **x** into a fixed high-dimension vector $\mathbf{z} \in \mathbb{R}^T$, which can be taken as the distribution of document **x** on T topics.

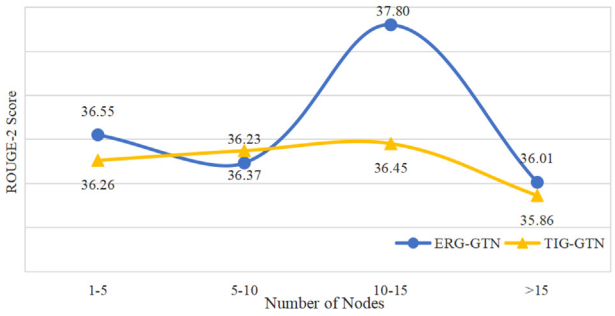

Fig. 5. Results of different number of graph nodes on LPO-news test set.

- ETM [6] is an improved neural topic model based on NVDM, which is designed to tackle the large and heavy-tailed vocabulary problem by generating the topic distribution in the embedding spaces.

To construct the vocabulary, we filter out the stop words, and the words appeared less than 5 times. The extracted elements are added to the vocabulary to ensure their topic distribution can be generated. Finally, we get a vocabulary with 41,519 words. For all topic models, the number of topics T is set to 200. We train the LDA model using the genism script [9] with 1000 iterations, the NVDM and ETM model[10] for 100 epochs with batch_size = 512. Specifically, for the ETM model, the embedding size is set to 300.

To introduce the TIG graph obtained by different topic models, we list the examples in Fig. 6 of *Zhang Yingying Case*, which is described in Fig. 2. We first notice that the topic words generated by different topic models are quite different, reflecting the ability of these topic models. ETM model generates highly correlated topic words such as *kidnapped, jury, FBI* etc., while NVDM and LDA tend to generate mixed and noisy topic words. For example, the LDA model mixes the Zhang Yingying Case and air hostess murder case, which have similar case elements, e.g., female victim, kidnapped, murder. It can also be summarized from Fig. 6 that, in contrast to the LDA-based method, the neural topic model (NVDM and ETM) can generate more intensive TIG graph. We attribute the possible reason that the neural topic model utilizes a deep neural network to model topic information, capturing deeper topic-related information rather than shallow word co-occurrence information. Moreover, We draw the following conclusions: LDA model tends to omit important correlation information (e.g., node ② and ③ should be linked to reflect the murder Christensen is from Illinois university), while NVDM model tends to build wrong correlation (e.g., node ① and ④ are connected, which may be wrongly interpreted as Zhang is an American).

We further discuss whether the topic model is essential for the summarization task. The results of the TIG-GTN model with different topic models are presented in Table 5. First, we observe that the LDA and ETM model achieve approximate performance but significantly outperform the NVDM-based model. We conjecture that the NVDM-based TIG graph may contain many noisy links, which leads to an obvious degradation. In contrast, although the LDA model may ignore some links, resulting in a coarser event skeleton frame, it can still capture the source document's important information. Therefore, there is only a slight decline compared with the ETM model.

6.2.5. The impact of different similarity methods for TIG

In Section 3, we introduce the construction of topic interaction graph (TIG) based on several different topic models, where the topic models are responsible for generating the topic representations of the source documents, then the similarity methods are introduced to build the relations by estimating the similarity based on the topic representations. Hence, in this section, we investigate the impact of different similarity methods on the model's performance.

In detail, we present 5 common similarity calculation methods, including distance-based Euclidean distance and Manhattan distance and correlation-based Cosine Similarity, Jaccard Coefficient and Pearson Coefficient. We list the results of the TIG-GTN model with different similarity calculation methods on the LPO-news corpus in Table 6, where the topic model is ETM. As seen from the table, the results obtained by different similarity calculation methods show slight divergence. Moreover, the correlation-based methods (the bottom part in Table 6) achieve better performance than the distance-based methods. The possible reason is that the correlation-based methods are not sensitive to the value of the topic vector, so they can better model the correlations between topic representations.

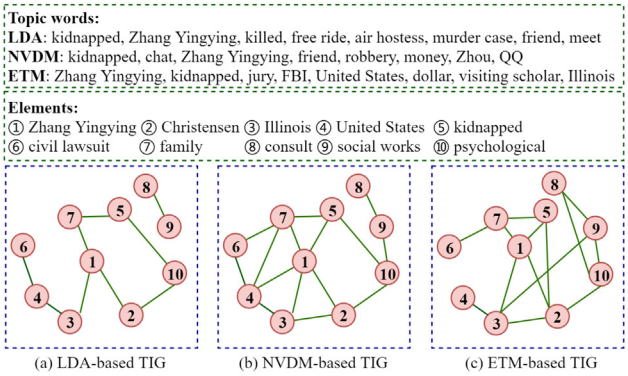

Fig. 6. Examples of TIG graph based on different topic models.

Table 5
Results of TIG-GTN on LPO-news with different topic models.

Topic Model	RG-1	RG-2	RG-L
LDA	43.96	36.24	41.50
NVDM	42.70	34.88	40.23
ETM	44.11	36.86	42.15

Table 6
Results of TIG-GTN with different similarity calculation methods on LPO-news test set.

Similarity Methods	RG-1	RG-2	RG-L
Euclidean Distance	43.27	35.84	41.66
Manhattan Distance	43.70	35.96	41.91
Cosine Similarity	44.11	36.86	42.15
Jaccard Coefficient	44.06	36.29	42.12
Pearson Coefficient	43.91	36.13	41.83

6.2.6. The impact of hyper-parameter γ

For TIG, the hyper-parameter γ is crucial to adjust the sparsity of the graph. Hence, we further investigate the TIG-GTN model's ability concerning different γ on the LPO-news test set, and the results are presented in Table 7. Note that the AvgEdges denotes the average number of the edges in the training set. In general, when $\gamma = 0.3$, the TIG-GTN model achieves the best performance. In contrast, the model obtains severe drops when γ is set to 0.1. It means that the graph is too dense to capture the topic interaction information between the elements. Similar trends can be observed at $\gamma = 0.7$. In this case, the graph is very sparse, losing the necessary connections in the graph.

Table 7
Results of different γ for the TIG-GTN model on LPO-news test set.

γ	AvgEdges	RG-1	RG-2	RG-L
0.1	26.35	37.28	30.66	35.95
0.3	18.40	43.96	36.24	41.50
0.5	10.85	42.11	35.03	40.34
0.7	6.44	39.62	34.27	38.61

[9] https://radimrehurek.com/gensim/models/ldamodel.html
[10] https://github.com/zll17/Neural_Topic_Models.

6.2.7. The impact of the layers of the model

In this paper, we use a standard Transformer configuration with a 6-layers encoder (including sequential encoder and graph encoder) and decoder, resulting in about 54M parameters. To verify the model's performance with different parameter scales, we conduct detailed experiments on the ERG-GTN model with varying layers on both internal test set and standard test set and present the results in Table 8. Specifically, the internal test set contains 1,000 samples, which are randomly extracted from the training data. As shown in Table 8, we first observe that when the model is very shallow (only 1 or 2 layers), the performance is poor both in the training set and test set. We assume that the shallow models do not have sufficient ability to fit the current summarization task data. We can also see that with increasing the model's layer, the model's performance has no significant improvement. The possible reason is that the scale of the training set is small (only 121,853) and the data pattern is simple (only 70 words for the source document and 12 words for the summary), so it not difficult for the summarization model to learn the mapping function between the source document and target summary. However, when the model is very deep (>10), the model's performance degrades seriously due to gradient vanishes and overfitting, which are common and intractable problems in the deep networks.

6.2.8. Is it better to mix different graphs

In this paper, we propose two types of element graph to enhance the case-related news summarization task. The ERG graph is constructed by directly linking the elements using virtual nodes, capturing the document-level interaction between these elements. In contrast, the TIG graph is built by estimating the similarities of different elements based on the topic-level representation to capture the interaction of the elements. These two kinds of graphs are both useful for the summarization model. To further evaluate whether the combination of different graphs can achieve better performance, we mix the ERG and TIG to construct the unified graph (UG) model and introduce a Full Connection Graph model (FCG model) that treats the elements as a full connection graph. As shown in Table 9, the UG model does not achieve better performance by mixing these two different structures. We conjecture the possible reason is that simple integrating ERG and TIG may destroy the inherent structure of the graph. Moreover, if all elements are taken as a fully connected graph without considering the structural information, in that case, the model degenerates into a self-attention network and gains a significant drop compared with ERG and TIG. These observations also confirm our conjecture that it is useful for the model to understand the events by capturing the structural information between different elements.

Table 9
Results of TIG-GTN model with different mixed graphs.

Model	RG-1	RG-2	RG-L
ERG	44.28	36.64	43.15
TIG	43.96	36.24	41.50
UG	42.18	33.95	40.82
FCG	41.57	32.11	39.83

6.3. Human evaluation

To further evaluate the quality of the generated summary, we carry out a human evaluation focusing on three aspects: **faithfulness**, **informativeness**, and **fluency**. The faithfulness metric evaluates whether the model generates irrelevant or conflicting summaries with the source document; the informativeness metric measures whether the generated summary covers the source document's main aspect; the fluency metric measures whether the generated summary is grammatically correct and accords with the logic. In detail, we randomly sample 100 examples from the LPO-news test set and ask five raters to score the examples with 1∼5, where 1 indicates the worst and 5 is the best. Finally, we perform average operation across different raters to generate the final score.

As shown in Fig. 7, we first note that the ERG-GTN and TIG-GTN model not only perform much better than TF and TF-BERT model but also achieves an equivalent performance with the gold summary in all three metrics, which indicates the effectiveness of the element graph. For the metric of faithfulness, the ERG-GTN model receives a high score of 3.76, which outperforms other models by a large margin, demonstrating that the model can learn the source document's factual representation introducing the element graph to avoid generating summaries that conflict with the source document. In the informativeness metric, the ERG-GTN model also achieves the best performance over the other baselines. It indicates that our model can capture the source document's main topic instead of referring to the minor details or high-frequency terms, which leads to more coherence and concise summary. We also notice that our ERG-GTN model gains a slight increment than the gold summary in informativeness. The possible reason is that the gold summary may contain some noise, while our model produces higher quality summaries. For the fluency metric, our TIG-GTN

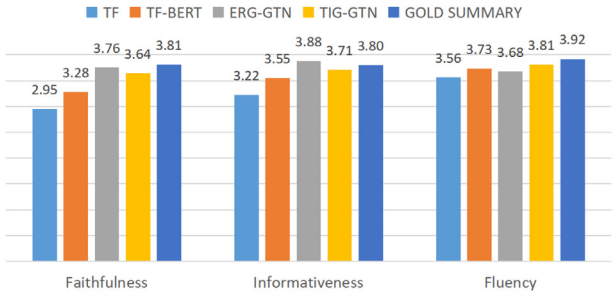

Fig. 7. Results of human evaluation.

Table 8
Results of ERG-GTN model with different layers on LPO-news training set and test set. The Δ denotes the deviation of different layer models compared to the 6-layer transformer model under the ROUGE-1 metric.

Layer Number	Parameters	Internal Test				Test			
		RG-1	RG-2	RG-L	Δ	RG-1	RG-2	RG-L	Δ
1	14.13M	47.82	34.66	46.91	−**3.75**	40.88	32.03	40.08	−**3.40**
2	22.28M	50.06	37.04	48.84	−**1.51**	42.29	34.87	41.66	−**1.99**
3	30.43M	51.51	38.12	50.06	−**0.06**	43.81	35.96	43.28	−**0.47**
4	38.58M	50.64	37.47	49.39	−**0.93**	44.35	36.58	43.23	**+0.07**
5	46.73M	52.02	38.56	50.77	**+0.45**	44.25	36.60	43.28	−**0.03**
6	54.88M	51.57	38.64	50.15	-	44.28	36.64	43.15	-
10	87.48M	50.02	37.36	48.67	−**1.55**	42.03	34.17	41.66	−**2.25**

model acquires a notable improvement from the TF model, and comparable results with the TF-BERT model indicate the BERT model is useful for the model to understand the source document. However, we also observe that the ERG-GTN model achieves a low result of 3.68, which is lower than the 3.81 of the TIG-GTN model. To find out why, we check out the outputs of the ERG-TGN model and notice that there are some repetitive words, which seriously affect the readability of the summary. We hypothesis that the ERG model introduces some external virtual nodes to connect the graph, which is useless for the summary generation. In a word, extensive human evaluation further validates the effectiveness of our proposed element-graph augmented model.

Source Document	怀疑丈夫与同村的李老太有染为报复情敌半年中##岁的胡老太竟然#次深夜放火焚烧对方房屋以泄愤恨。近日陕西省人民检察院以放火罪对胡老太提起公诉 (Suspected that her husband had an affair with Mrs. Li , who is from the same village, Mrs. Hu, ## years old, set fire to Li's house # times in half a year to vent her anger. Recently, the Shanxi Provincial People's Procuratorate initiated a public prosecution against Mrs. Hu for arson.)
Gold Summary	六旬老太疑丈夫出轨五次深夜放火情敌房屋 (The 60-year-old woman suspected that her husband had cheated, and set fire to the enemy's house five times at night.)
TF-BERT	男子怀疑妻子与同村人有染为报复情敌砍伤老太 (The man suspected that his wife had an affair with the villagers, and cut down the old lady for revenge.)
ERG-GTN	陕西老太疑丈夫与同村人有染放火焚烧对方房屋 (The old lady of Shanxi suspected that her husband had an affair with the villagers and burned her house)
Source Document	美国艾奥瓦大学近百名中国留学生因涉嫌学术欺诈面临开除的消息近日从国外社交平台传到国内，引发关注。涉事留学生所修的是在线课程，缘于觉得这门课简单 (The news that nearly 100 Chinese students from the University of Iowa are facing dismissal for suspected academic fraud recently spread from foreign to China. The international students involved took an online course because they thought it was simple.)
Gold Summary	近百名中国留美学生被曝涉嫌学术欺诈面临开除 (Nearly 100 Chinese students studying in the United States have been dismissal on suspicion of academic fraud.)
TF-BERT	中国留学生涉嫌学术欺诈所修在线课程 (Chinese students took an online course and suspected of involving academic fraud.)
ERG-GTN	中国留学生因学术欺诈面临开除 (Chinese students faced dismissal for academic fraud)

Fig. 8. Summaries generated by different models.

6.4. Case study

We present two examples for better understanding of the performance of our proposed model. As shown in Fig. 8, the first example mainly expresses that "an old woman who suspected that her husband was cheating and burned her rival's house", while the TF-BERT model mistakenly generates that "the man suspected his wife of cheating and cut down her rival", which conflicts with the source document. We infer that the model has a deviation in understanding the source document and can not clearly distinguish the relationship of the persons involved in this event. In contrast, our ERG-GTN model produces the correct summary by utilizing the element graph as the source document's structural representation. In the second case, although the TF-BERT generates the accurate summary, it focuses on the secondary point of "took an online course", thus hurting the summary's conciseness. But our model introduces the graph to capture the main aspect of the source document and yields an informative and concise summary.

Furthermore, we present some extra examples generated by the ERG model to analyze our proposed model and give some potential direction for further improvement. As shown in Fig. 9, the first two examples show that the ERG model can generate better summaries with appropriate replacements or additions during the summary generation process. For instance, the first example replaces the *good-looking suspect* with *Li Qingwu*, and the second example adds *secretary of the Hebei Province* as attribute before *Zhou Benshun*, which make the generated summary easy to understand. On the contrary, the latter two examples generate wrong summaries due to inappropriate replacements and additions raised by the element graph. As the third example, the *director of transportation bureau* is identified as the subject of the suicide case, and the Baotou in the last example is mistakenly identified as the *city* of the suspect. We argue the possible reason is that the ERG may contain some noisy correlations, leading to the wrong summarizes. To alleviate this problem, we propose several possible improvement directions: (1) using more data (such as reader comments or history information of the case) to build accurate element relationships; (2) using more power encoder (such as encoding the relation into the node representation) to generate better representation of the graph;

Appropriate Replacements	Source Document	近日，海南警方公布的通缉涉黑涉恶在逃人员通告中，一位名叫李庆武的嫌疑人，因高颜值照片意外走红。李庆武涉嫌参加黑社会性质组织罪，已于通缉令发布#天后投案自首 (Recently, a suspect named Li Qingwu, who was in Hainan police's notice involved in crimes, unexpected became a popular due to his good-looking photo. Li Qingwu was suspected to be involved in the crime of underworld nature organization. He surrendered himself to the police # days after the arrest warrant was released.)
	TF-BERT	高颜值通缉犯走红网络警方涉嫌参加黑社会已投案自首 (Good looking suspect became popular and suspected of participating underworld nature organization, and has surrendered himself to the police.)
	ERG-GTN	海南警方通缉涉黑涉恶逃犯李庆武已自首 (Li Qingwu, a suspected wanted by Hainan police for involvement in crimes has surrendered himself to the police.)
Appropriate Additions	Source Document	据中央组织部有关负责人证实，河北省委书记、省人大常委会主任周本顺涉嫌严重违纪违法，中央已决定免去其领导职务，现正在按程序办理 (According to the relevant person in charge of the Organization Department of the CPC Central Committee, Zhou Benshun, secretary of the Hebei Provincial Committee and director of the Standing Committee of the Provincial People's Congress, was suspected of serious violations of discipline and law. The CPC Central Committee has decided to remove him from his leadership position, and is now proceeding according to the procedure.)
	TF-BERT	周本顺被免职 (Zhou Benshun was removed from office.)
	ERG-GTN	河北省委书记周本顺被免职 (Zhou Benshun, Secretary of Hebei provincial Party committee, was removed from office)
Inappropriate Replacements	Source Document	河南永城女车主不堪罚款服毒自杀事件最新进展今日，永城市委外宣办通报了处理结果市交通运输局运管局局长、分管副局长给予停职检查处理；对运管局运政大队大队长给予免职处理 (The latest progress of female car owner's suicide case in Yongcheng, Henan Province. Today, the Yongcheng city committee reported the disposal results: The director and deputy director in charge of the Municipal Transportation Bureau were suspended from duty. The chief of the Transportation Administration Brigade of the Transportation Administration Bureau will be removed from office.)
	TF-BERT	河南回应女车主服毒:运管局长等被停职免职 (Henan responded to the case of female car owners taking drugs: the director of Municipal Transportation Bureau is suspended from duty.)
	ERG-GTN	河南永城交通局局长不堪罚款自杀事件#人停职 (The director of Henan Yongcheng Transportation Bureau committed suicide for the fine, and # people were suspended from their duties.)
Inappropriate Additions	Source Document	高校教授都林因参与包头市政府高速公路建设遭受#年冤狱。#月##日，他收到包头市中院##余万国家赔偿决定书 (Du Lin, a university professor, was wrongly jailed for # years for his participation in the construction of Baotou municipal government expressway. On #, he received a decision of more than # million state compensation from Baotou City Intermediate People's Court.)
	TF-BERT	教授因参与政府高速公路建设蹲七年冤狱法院赔##万 (The professor was sentenced to seven years of unjust imprisonment for participating in the construction of government expressways, and the court compensated # million yuan.)
	ERG-GTN	包头一教授涉贪冤狱##年获###万国家赔偿 (A professor in Baotou was involved in corruption and unjust imprisonment # # years, and received # # # million state compensation)

Fig. 9. The examples generated by ERG models. The words marked blue are appropriate replacements or additions, while the words marked red are inappropriate replacements or additions.

(3) using more effective fusion method to combine the graph and sequential encoder.

7. Conclusion

This paper introduces an element-graph enhanced abstractive summarization model utilizing explicit structural information to produce more practical summaries for event-based news such as legal public opinion news. Our proposed model benefits from the global event-plan information captured from the element graph, thus focusing on the source document's main aspects and avoiding factual errors. Extensive experiments demonstrate that our model can generate a more informative, coherent, and faithful summary. In the future, we would like to explore better ways to construct the knowledge graphs based on the source document or utilize external knowledge obtained from larger-scale open-domain KG to improve the news domain summarization task consistently.

CRediT authorship contribution statement

Yuxin Huang: Writing - original draft, Writing - review & editing, Methodology. **Zhengtao Yu:** Conceptualization, Methodology, Supervision, Funding acquisition. **Junjun Guo:** Software, Visualization. **Yan Xiang:** Methodology. **Yantuan Xian:** Software, Validation.

CRediT authorship contribution statement

Yuxin Huang: Writing - original draft, Writing - review & editing, Methodology. **Zhengtao Yu:** Conceptualization, Methodology, Supervision, Funding acquisition. **Junjun Guo:** Software, Visualization. **Yan Xiang:** Methodology. **Yantuan Xian:** Software, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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 [Grant Nos. 2018YFC0830105, 2018YFC0830101, 2018YFC0830100]; the National Natural Science Foundation of China [Grant Nos. 61972186, 61762056, 61472168]; the Yunnan provincial major science and technology special plan projects [Grant No. 202002AD080001]; General projects of basic research in Yunnan Province [Grant No. 202001AT070047].

References

[1] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014, arXiv preprint arXiv:1409.0473..
[2] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, K. Sima'an, Graph convolutional encoders for syntax-aware neural machine translation, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 1957–1967. https://www.aclweb.org/anthology/D17-1209, 10.18653/v1/D17-1209..
[3] D. Beck, G. Haffari, T. Cohn, Graph-to-sequence learning using gated graph neural networks, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 273–283. https://www.aclweb.org/anthology/P18-1026, 10.18653/v1/P18-1026..
[4] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.
[5] M. Damonte, S.B. Cohen, Structural neural encoders for AMR-to-text generation, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 3649–3658. https://www.aclweb.org/anthology/N19-1366, 10.18653/v1/N19-1366..
[6] A.B. Dieng, F.J.R. Ruiz, D.M. Blei, Topic modeling in embedding spaces, Transactions of the Association for Computational Linguistics 8 (2020) 439–453. URL: https://www.aclweb.org/anthology/2020.tacl-1.29, 10.1162/tacl_a_00325.
[7] Y. Dong, A. Romascanu, J.C. Cheung, Hiporank: Incorporating hierarchical and positional information into graph-based unsupervised long document extractive summarization, 2020, arXiv preprint arXiv:2005.00513..
[8] P. Fernandes, M. Allamanis, M. Brockschmidt, Structured neural summarization, 2018, arXiv preprint arXiv:1811.01824..
[9] C. Gardent, A. Shimorina, S. Narayan, L. Perez-Beltrachini, The WebNLG challenge: Generating text from RDF data, in: Proceedings of the 10th International Conference on Natural Language Generation, Association for Computational Linguistics, Santiago de Compostela, Spain, 2017, pp. 124–133. https://www.aclweb.org/anthology/W17-3518, 10.18653/v1/W17-3518..
[10] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, Convolutional sequence to sequence learning, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 1243–1252..
[11] S. Gehrmann, Y. Deng, A. Rush, Bottom-up abstractive summarization, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4098–4109. https://www.aclweb.org/anthology/D18-1443, 10.18653/v1/D18-1443..
[12] S. Gupta, S. Gupta, Abstractive summarization: An overview of the state of the art, Expert Systems with Applications 121 (2019) 49–65.
[13] K.M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, in: Advances in Neural Information Processing Systems, 2015, pp. 1693–1701.
[14] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp, in: International Conference on Machine Learning, 2019, pp. 2790–2799.
[15] L. Huang, L. Wu, L. Wang, Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, Online, pp. 5094–5107. URL: https://www.aclweb.org/anthology/2020.acl-main.457.
[16] Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980..
[17] Kingma, D.P., Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114..
[18] Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907..
[19] Koncel-Kedziorski, R., Bekal, D., Luan, Y., Lapata, M., Hajishirzi, H., 2019. Text Generation from Knowledge Graphs with Graph Transformers, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota. pp. 2284–2293. https://www.aclweb.org/anthology/N19-1238, 10.18653/v1/N19-1238..
[20] Lebret, R., Grangier, D., Auli, M., 2016. Neural text generation from structured data with application to the biography domain, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas. pp. 1203–1213. https://www.aclweb.org/anthology/D16-1128, 10.18653/v1/D16-1128..
[21] C. Li, W. Xu, S. Li, S. Gao, Guiding generation for abstractive text summarization based on key information guide network, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 55–60. https://www.aclweb.org/anthology/N18-2009, 10.18653/v1/N18-2009..
[22] H. Li, J. Zhu, J. Zhang, C. Zong, X. He, Keywords-guided abstractive sentence summarization, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 8196–8203.
[23] W. Li, X. Xiao, J. Liu, H. Wu, H. Wang, J. Du, Leveraging graph to improve abstractive multi-document summarization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, Online, pp. 6232–6243. URL: https://www.aclweb.org/anthology/2020.acl-main.555.
[24] W. Li, J. Xu, Y. He, S. Yan, Y. Wu, X. Sun, Coherent comments generation for Chinese articles with a graph-to-sequence model, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4843–4852. https://www.aclweb.org/anthology/P19-1479, 10.18653/v1/P19-1479..
[25] Y. Li, D. Tarlow, M. Brockschmidt, R. Zemel, Gated graph sequence neural networks, 2015, arXiv preprint arXiv:1511.05493..
[26] C.Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics,

Barcelona, Spain, 2004, pp. 74–81. URL https://www.aclweb.org/anthology/W04-1013.
[27] J. Lin, X. Sun, S. Ma, Q. Su, Global encoding for abstractive summarization, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 163–169. https://www.aclweb.org/anthology/P18-2027, 10.18653/v1/P18-2027..
[28] C. Ma, W.E. Zhang, M. Guo, H. Wang, Q.Z. Sheng, Multi-document summarization via deep learning techniques: A survey, 2020, arXiv preprint arXiv:2011.04843.
[29] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 55–60. URL: https://www.aclweb.org/anthology/P14-5010, 10.3115/v1/P14-5010.
[30] D. Marcheggiani, J. Bastings, I. Titov, Exploiting semantics in neural machine translation with graph convolutional networks, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 486–492. https://www.aclweb.org/anthology/N18-2078, 10.18653/v1/N18-2078..
[31] Y. Miao, E. Grefenstette, P. Blunsom, Discovering discrete latent topics with neural variational inference, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 2410–2419..
[32] R. Mihalcea, P. Tarau, TextRank: Bringing order into text, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 404–411. https://www.aclweb.org/anthology/W04-3252..
[33] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, B. Xiang, Abstractive text summarization using sequence-to-sequence RNNs and beyond, in: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 280–290. https://www.aclweb.org/anthology/K16-1028, 10.18653/v1/K16-1028..
[34] S. Narayan, S.B. Cohen, M. Lapata, Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1797–1807. https://www.aclweb.org/anthology/D18-1206, 10.18653/v1/D18-1206..
[35] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, M. Auli, fairseq: A fast, extensible toolkit for sequence modeling, in: Proceedings of NAACL-HLT 2019: Demonstrations, 2019.
[36] A. See, P.J. Liu, C.D. Manning, 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), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1073–1083. https://www.aclweb.org/anthology/P17-1099, 10.18653/v1/P17-1099..
[37] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1715–1725. https://www.aclweb.org/anthology/P16-1162, 10.18653/v1/P16-1162..
[38] L. Song, D. Gildea, Y. Zhang, Z. Wang, J. Su, Semantic neural machine translation using AMR. Transactions of the Association for Computational Linguistics 7 (2019) 19–31. URL: https://www.aclweb.org/anthology/Q19-1002, 10.1162/tacl_a_00252.
[39] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
[40] K.S. Tai, R. Socher, C.D. Manning, 2015. Improved semantic representations from tree-structured long short-term memory networks, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China. pp. 1556–1566. URL https://www.aclweb.org/anthology/P15-1150, 10.3115/v1/P15-1150.
[41] B.D. Trisedya, J. Qi, R. Zhang, W. Wang, GTR-LSTM: A triple encoder for sentence generation from RDF data, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1627–1637. https://www.aclweb.org/anthology/P18-1151, 10.18653/v1/P18-1151..
[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph Attention Networks. International Conference on Learning Representations, 2018, https://openreview.net/forum?id=rJXMpikCZ. accepted as poster..
[44] X. Wan, An exploration of document impact on graph-based multi-document summarization, in: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Honolulu, Hawaii, 2008, pp. 755–762. URL: https://www.aclweb.org/anthology/D08-1079.
[45] D. Wang, P. Liu, Y. Zheng, X. Qiu, X. Huang, Heterogeneous graph neural networks for extractive document summarization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, Online, pp. 6209–6219. URL: https://www.aclweb.org/anthology/2020.acl-main.553.
[46] L. Wang, J. Yao, Y. Tao, L. Zhong, W. Liu, Q. Du, A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 4453–4460.
[47] H. Xu, Y. Wang, K. Han, B. Ma, J. Chen, X. Li, Selective attention encoders by syntactic graph convolutional networks for document summarization, in: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 8219–8223.
[48] W. Xu, C. Li, M. Lee, C. Zhang, Multi-task learning for abstractive text summarization with key information guide network, EURASIP Journal on Advances in Signal Processing 2020 (2020) 1–11.
[49] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, D. Radev, Graph-based neural multi-document summarization, in: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 452–462. https://www.aclweb.org/anthology/K17-1045, 10.18653/v1/K17-1045..
[50] S. Yun, M. Jeong, R. Kim, J. Kang, H.J. Kim, Graph transformer networks, in: Advances in Neural Information Processing Systems, 2019, pp. 11983–11993.
[51] T. Zhang, V. Kishore, F. Wu, K.Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with bert, 2019, arXiv preprint arXiv:1904.09675..
[52] Y. Zhang, Q. Liu, L. Song, Sentence-state LSTM for text representation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 317–327. https://www.aclweb.org/anthology/P18-1030, 10.18653/v1/P18-1030..
[53] Q. Zhou, N. Yang, F. Wei, M. Zhou, Selective encoding for abstractive sentence summarization, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1095–1104. https://www.aclweb.org/anthology/P17-1101, 10.18653/v1/P17-1101..
[54] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, T. Liu, Incorporating bert into neural machine translation, in: International Conference on Learning Representations, 2019.

Yuxin Huang was born in 1983 year. He is a Ph.D. candidate in computer science at Kunming university of Science and Technology. His research interests include natural language processing, text summarization, machine translation etc.

Zhengtao Yu was born in 1970. He received the Ph.D. degree in computer application technology from Beijing Institute of Technology in 2005. Now he is a professor and Ph.D. supervisor at Kunming University of Science and Technology, and the director of Yunnan Key Laboratory of Artificial Intelligence. His research interests include natural language processing, machine translation and information retrieval, etc.

Junjun Guo was born in 1988. He received the Ph.D. degree from Xi'an Jiao Tong University in 2016. Now He is a lecturer at Kunming University of Science and Technology. His research interests include natural language processing, machine translation etc.

Yan Xiang was born in 1979. She received the M.S. degree from Wuhan University in 2001. She is currently a Ph.D. candidate in computer science at Kunming University of Science and Technology. Her research interests include medical image processing, natural language processing, sentiment classification, and text mining etc.

Yantuan Xian was born in 1981. He received the M.S. degree from Shenyang Institute of Automation, Chinese Academy of Science in 2007. Now He is an associate professor at Kunming University of Science and Technology. His research interests include natural language processing, machine translation, text mining etc.