



# A Graph-to-Sequence Learning Framework for Summarizing Opinionated Texts

Penghui Wei , Jiahao Zhao, and Wenji Mao 

**Abstract**—There is a great need for effective summarization methods to absorb the key points of large amounts of opinions expressed on the Web. In this paper, we study the problem of opinionated text summarization, which aims to generate a coherent summary for a set of opinionated texts towards a specific topic (e.g., a movie or a controversial issue). The main characteristic of this problem is that the input set contains an arbitrary number of texts, which brings about redundant opinions and useless texts. Further, informative opinions to be summarized are scattered over different opinionated texts, thus it is vital to avoid focusing only on partial opinions. However, previous work can not tackle the above two issues effectively. To address such issues, we propose a two-stage graph-to-sequence learning framework for summarizing opinionated texts. The first stage selects summary-worthy texts from all input opinionated texts, and we construct an opinion relation graph to help estimate salience via exploiting the relationships among the input texts. Given the selected texts, the second stage generates an opinion summary via a maximal marginal relevance guided graph-to-sequence model, which gives consideration to both salient and non-redundant opinions. Experimental results on two benchmark datasets show that our framework outperforms the existing state-of-the-art methods. Human evaluation further verifies that our framework can generate more informative and compact opinion summaries than previous methods.

**Index Terms**—Opinion summary generation, Graph neural networks, Graph-to-sequence learning, Maximal marginal relevance.

## I. INTRODUCTION

WITH the widespread use of social media, people are habituated to expressing opinions towards topical issues and exchanging their views with others. Massive amounts of opinionated texts accumulate rapidly on the Web. Discovering the key points of diverse opinions can help us better understand public concerns towards a certain topic (e.g., a product, a social event or a controversial issue), facilitating decision-making,

Manuscript received October 18, 2020; revised March 5, 2021; accepted March 22, 2021. Date of publication April 9, 2021; date of current version May 10, 2021. This work was supported in part by NSFC under Grants #11832001 and #71621002, in part by the Ministry of Science and Technology of China under Grants #2020AAA0108401 and #2020AAA0108405, and in part by Beijing Nova Program Z201100006820085 from Beijing Municipal Science and Technology Commission. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sakrani Sakti. (Corresponding author: Wenji Mao.)

The authors are with the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China, and School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China (e-mail: weipenghui2016@ia.ac.cn; zhaojiahao2019@ia.ac.cn; wenji.mao@ia.ac.cn).

Digital Object Identifier 10.1109/TASLP.2021.3071667

---

Movie: *Mission: Impossible — Ghost Protocol*

Review 1: A *fast-paced*, globe-trotting adventure that offers blasts, brawls and buildings of equally spectacular scale.

Review 2: Take off thinking cap and munch on *extra buttery popcorn*. It is candy for the brain, a straightforward action flick...

Review 3: Bird has done a *stylish* and involving job here, turning in an entertaining production that's got considerable visual flair...

Review 4: With a fun group of characters and cool *set pieces* that take advantage of the IMAX format, it always manages to...

Review 5: Mission: Impossible - Ghost Protocol is *top-notch popcorn entertainment*, chock-full of dazzling stunts and heroic...

Consensus: *Stylish, fast-paced*, and loaded with gripping *set pieces*, the Mission: Impossible is *big-budget popcorn entertainment* that really works.

---

(a) Reviews about the movie “Mission: Impossible – Ghost Protocol”, and an opinion consensus as a summary.

---

Controversial Issue: *Generic Drugs*

Argument 1: The denial of *the right to produce* or acquire generic drugs is effectively a death sentence to people in *developing countries*.

Argument 2: There is no ethical justification to allow pharmaceutical companies to charge artificially high prices for drugs that *save lives*.

Argument 3: With generic drugs freely available on the market, the access to such drugs would be facilitated far more readily and cheaply.

Claim: *Allowing production* of generic drugs *saves lives*, particularly in the *developing world*.

---

(b) Arguments on the controversial issue “Generic Drugs”, and a central claim as a summary.

Fig. 1. Two types of opinionated text: (a) reviews, and (b) arguments on a controversial topic.

policy evaluation and risk management for individual, public and private sectors. However, digesting opinionated information is an overwhelming task, because user-generated contents are mixed with miscellaneous opinions, and the discovery of interested contents needs intensive manual labor [1]. Thus, there is a great need for effective summarization methods that can efficiently absorb the key points of large amounts of opinions expressed on the Web and automatically generate informative and concise opinion summaries.

*Opinionated text summarization* aims to generate a coherent summary for a set of opinionated texts towards a specific topic [2]. Fig. 1 shows examples for two types of opinionated text studied in this work, i.e., reviews and arguments. Five out of 211 reviews for a movie with an opinion consensus of them are illustrated in Fig. 1(a). Moreover, three out of 14 arguments that support a controversial topic with a central claim derived from them are shown in Fig. 1(b). In this scenario, the task is formulated as claim generation [2], which can be the next step after stance classification [3]–[5] and argument clustering [6]. One characteristic of opinionated text summarization task is that the number of input texts can be arbitrary and might be large, which brings about redundant opinions and useless texts.

Furthermore, informative opinions to be summarized are scattered over multiple different opinionated texts. These characteristics indicate that an opinionated text summarization system should be capable of identifying important texts and filtering out noisy ones, as well as paying attention to diverse opinionated texts for generating informative summaries.

As extractive-based summarization methods are less effective for the above characteristics (especially the second one), the existing state-of-the-art methods of opinionated text summarization [2], [7] are abstractive-based methods, which employ the encoder-decoder neural network [8] widely used in sequence-to-sequence learning tasks. Two representative methods [2], [7] have been proposed and achieve better performance than other existing extractive- and abstractive-based methods [9]–[11]. Wang and Ling [2] proposed a two-stage method for opinionated text summarization, in which the first stage computes the salience scores of input texts using a ridge regression model, and then the second stage takes a subset of all texts as the input (e.g., concatenating the top-5 texts to form a long sequence) to generate an opinion summary with an attention-based encoder-decoder model. Amplayo and Lapata [7] also presented a two-stage method, where the first stage is condensing that learns vector representations for all input texts by an auto-encoder model, and then an attention-based fusion mechanism is used to aggregate them into a single vector representation, which is fed into an RNN decoder for generating an opinion summary at the second stage.

Although the above methods have achieved the state-of-the-art performance, they are confronted with two issues. First, to identify important texts and filter out noisy ones from multiple input texts, previous studies [10], [12] have shown that considering the relationships among texts can improve the performance of summarization, yet the two methods [2], [7] do not explicitly utilize the relationships among input opinionated texts during their first stage: the method in [2] independently estimates the salience score of each opinionated text, and the method in [7] also does not consider the relationships of input texts during learning vector representation for each text. Second, informative opinions are dispersed in multiple texts, and it is vital to avoid focusing on only partial opinions and generating repetitive opinions. However, these two methods employ the classic RNN decoder to generate summaries, and such process lacks key information that can guide the decoder which opinions have been generated and which ones should be generated at subsequent time steps.

To address the above issues, in this paper, we propose a two-stage graph-to-sequence learning framework for opinionated text summarization. At the first stage, given a set of opinionated texts, we construct an opinion relation graph to explicitly exploit the relationships among the opinionated texts, and adopt a graph neural network on such graph to aggregate and synthesize information for estimating salience scores of all texts. At the second stage, we employ graph-to-sequence learning to generate opinion summaries. A graph neural network encoder learns the text representations of opinionated texts selected by the first stage, and then a maximal marginal relevance (MMR) [13] guided decoder is adopted to generate an informative and coherent summary. The integrated MMR mechanism can guide

the decoder giving consideration to both salient opinions and non-redundant opinions with marginal relevance scores, and refraining from paying attention to repetitive opinions that have been generated at previous time steps. We conduct experiments on two benchmark datasets about movie reviews and arguments on controversial issues, and results show the effectiveness of our proposed framework.

The main contributions of our work are as follows.

- We propose a two-stage graph-to-sequence learning framework, which represents the set of input texts as a graph structure to explicitly model the relationships among them for improving opinionated text summarization.
- To generate more informative summaries and avoid focusing on only partial opinions from the input texts, we develop an MMR-guided decoder to refine the summary generation process.
- Experimental results on two benchmark datasets verify that our proposed framework outperforms the existing state-of-the-art methods. Human evaluation further shows that our framework generates more informative and compact opinion summaries than previous methods.

## II. RELATED WORK

The focus of our work is *opinionated text summarization*, which aims to build a coherent summary for a set of opinionated or argumentative texts [1], [2], [7], [9], [14], [15], or a single but long one [16]–[21]. The most related studies to our work are [1], [2], [7], [9]. Ganesan *et al.* [9] developed an unsupervised method to produce opinion summary by using a graph, where each node is a word and directed edges represents text structure, and the process of summarization is to find appropriate path in the graph based on large amounts of redundant opinions. Jang and Allan [1] studied the problem of summarizing tweets on Twitter for two-sided stances of controversial issues. They formulated this as a ranking task and proposed to extract representative tweets as stance summary. Wang and Ling [2] proposed a two-stage abstractive-based method that consists of a ridge regression model for estimating salience scores of opinionated texts and an attentive encoder-decoder for summary generation. Amplayo and Lapata [7] improved the first stage by an auto-encoder that learns to represent all input texts, and an attention layer than fuses them to generate opinion summaries with RNN decoder. Their method achieves the state-of-the-art performance for opinionated text summarization.

Another research line is *aspect-based opinion summarization* that aims to form summaries for aspects of a specific entity [22]–[32]. Early work [22], [23] treated opinion summary as a structured list containing aspects and positive/negative reviews towards each aspect. Follow-up work [24]–[28] performed extractive summarization that considers to extract representative reviews to form the opinion summary of each aspect. Recent work [29]–[32] further employed methods for abstractive summarization to generate aspect-aware opinion summaries. The work of Amplayo and Lapata [7] is the pioneering study that combines the above two types of opinion summarization, which can control the aspects of generated review summaries.

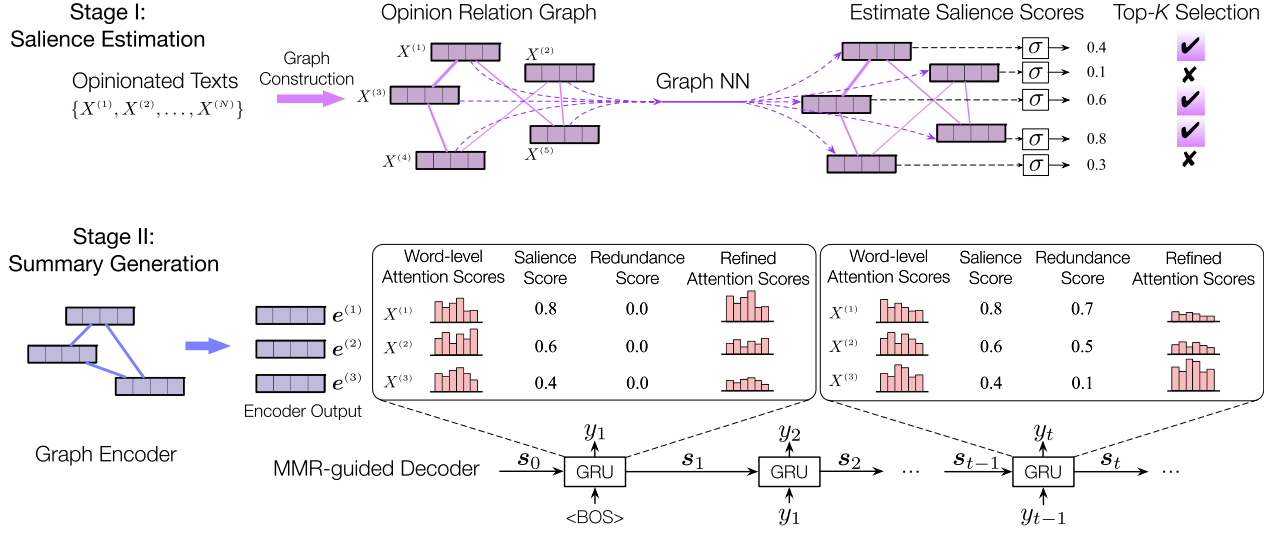


Fig. 2. Workflow of our proposed two-stage graph-to-sequence learning framework for opinionated text summarization.

Recent work on single-document summarization [33], [34] and multi-document summarization [35], [36] has paid attention to design an explicit content selection module for generating more meaningful summaries. These studies inspire us to incorporate MMR mechanism into the decoder for refining attention scores. Unlike previous studies, our work focuses on effective saliency estimation models by exploiting relationships among opinionated texts and giving consideration to both saliency and non-redundance during opinion summary generation.

### III. PROBLEM STATEMENT

We formulate the task of opinionated text summarization as follows. Given a set of  $N$  opinionated texts (towards a specific topic)  $\mathcal{X} = \{X^{(1)}, X^{(2)}, \dots, X^{(N)}\}$  and the corresponding opinion summary  $Y$ , where each text  $X^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_{l_i}^{(i)})$  is a word sequence and the summary  $Y = (y_1, y_2, \dots, y_{|Y|})$  is also a word sequence ( $l_i$  and  $|Y|$  denote the length of  $X^{(i)}$  and  $Y$ , respectively), the learning objective of opinionated text summarization models is to estimate the conditional probability  $p(Y | \mathcal{X})$ .

This task can be separated as two sub-tasks. The first one is *saliency estimation*, which aims to assign a saliency score for each text  $X^{(i)}$  in  $\mathcal{X}$  and output a subset  $\mathcal{X}^*$  containing top- $K$  texts with highest saliency scores ( $K < N$ ). The second one is *opinion summary generation* that estimates the conditional probability  $p(Y | \mathcal{X}^*)$ .

### IV. PROPOSED METHOD

We propose a two-stage graph-to-sequence learning framework for summarizing opinionated texts. The first stage performs saliency estimation, which selects a subset of summary-worthy texts from all input opinionated texts with a graph neural regression model, and the second stage generates a concise opinion summary given the selected texts with a graph-to-sequence network guided by an MMR mechanism. Fig. 2 gives an overview of our proposed framework.

#### A. Saliency Estimation With Graph Neural Networks

To leverage the relationships among  $N$  opinionated texts, we build an opinion relation graph for them, and employ a graph neural network regression model to estimate saliency scores of these texts.

1) *Constructing Opinion Relation Graph*: We represent the input  $\mathcal{X} = \{X^{(1)}, X^{(2)}, \dots, X^{(N)}\}$  as an opinion relation graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ . Each vertex  $v_i \in \mathcal{V}$  is a text  $X^{(i)}$ , and the edge  $e_{ij} \in \mathcal{E}$  between two vertices  $v_i$  and  $v_j$  is constructed based on the TF-IDF cosine similarity  $w_{ij}$  of  $X^{(i)}$  and  $X^{(j)}$ :

$$w_{ij} = \frac{\overrightarrow{\text{tfidf}}(X^{(i)})^\top \overrightarrow{\text{tfidf}}(X^{(j)})}{\|\overrightarrow{\text{tfidf}}(X^{(i)})\| \|\overrightarrow{\text{tfidf}}(X^{(j)})\|} \in [0, 1] \quad (1)$$

where  $\overrightarrow{\text{tfidf}}(X^{(i)}) \in \mathbb{R}_+^{|\mathcal{V}|}$  denotes the TF-IDF vector of a text  $X^{(i)}$ , and  $|\mathcal{V}|$  is the vocabulary size.

We consider two variants of opinion relation graph:

- **Unweighted graph**: All edges in the graph are unweighted, i.e., if the TF-IDF cosine similarity  $w_{ij} > 0$ , we construct an edge between  $X^{(i)}$  and  $X^{(j)}$ , otherwise there is no edge between the two vertices;
- **Weighted graph**: All edges in the graph are weighted, and the weight of an edge  $e_{ij}$  is equal to  $w_{ij}$ .

2) *Graph Neural Regression*: We adopt a multi-layer graph neural network (GNN) to learn text (vertex) representations that are used to estimate saliency scores for all texts, and we formulate this as a regression problem.

Our graph neural regression (GNR) model consists of an input encoding layer, a stack of GNN layers, and a regression layer that produces final saliency scores. We first use a GRU [37] network to encode each input text  $X^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_{l_i}^{(i)})$  in the set  $\mathcal{X}$ :

$$h_t^{(i)} = \text{GRU}(x_t^{(i)}, h_{t-1}^{(i)}), \quad t \in \{1, \dots, l_i\} \quad (2)$$

where  $x_t^{(i)}$  and  $h_t^{(i)}$  are the word embedding and hidden state of the  $t$ -th word  $x_t^{(i)}$  respectively. For the text  $X^{(i)}$ , we take the last

step hidden state as its text representation, denoted as  $\mathbf{h}^{(i)} \in \mathbb{R}^d$ . Thus we have  $\{\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \dots, \mathbf{h}^{(N)}\}$ .

We then stack multiple GNN layers [38], [39] operating on the opinion relation graph  $\mathcal{G}$ , which models the interaction among input texts by aggregating neighbors' features to effectively learn vertex representations.

Specifically, the first layer's input is text representations learned by GRU:  $\mathbf{H}_0 = (\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \dots, \mathbf{h}^{(N)})^\top \in \mathbb{R}^{N \times d}$ . For the  $m$ -th GNN layer ( $m \in \{1, \dots, M\}$ ), it takes vertex features  $\mathbf{H}_{m-1} \in \mathbb{R}^{N \times d}$  learned by previous layer as input, and outputs aggregated vertex features  $\mathbf{H}_m \in \mathbb{R}^{N \times d}$ :

$$\mathbf{H}_m = \tanh(\mathbf{A}_m \mathbf{H}_{m-1} \mathbf{W}_m + \mathbf{b}_m) \quad (3)$$

where the  $i$ -th row of  $\mathbf{H}_m$  is the feature representation of the text  $X^{(i)}$ ,  $\mathbf{W}_m$  and  $\mathbf{b}_m$  are learnable parameters, and each element  $[\mathbf{A}_m]_{ij}$  of the matrix  $\mathbf{A}_m \in \mathbb{R}^{N \times N}$  defines the aggregation strength of two texts  $X^{(i)}$  and  $X^{(j)}$ .

Different definitions for  $\mathbf{A}_m$  result in different graph neural networks, such as graph convolutional network [38] and graph attention network [39]. Next we detail our GNN-based regression models for opinion relation graph.

Recall that we have two variants of graphs, i.e., unweighted graph and weighted graph, thus we consider the following types of aggregation strength matrix  $\mathbf{A}_m$  to build three GNR models:

- i) Graph Convolutional Regression on Unweighted Graph (GCR-U): We first apply symmetric normalization to the original adjacency matrix of the opinion relation graph  $\mathcal{G}$ , and set  $\mathbf{A}_m$  in all GNN layers to the normalized adjacency matrix of the graph:

$$\mathbf{A}_m = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \quad (4)$$

where  $\mathbf{A} \in \{0, 1\}^{N \times N}$  denotes the original adjacency matrix of the graph, and  $\mathbf{D}$  denotes the degree matrix with  $[\mathbf{D}]_{jj} = \sum_k [\mathbf{A}]_{jk}$ .

- ii) Graph Convolutional Regression on Weighted Graph (GCR-W): It is similar to (i), but all edges are weighted, i.e.,  $[\mathbf{A}]_{ij} = w_{ij} \in \mathbb{R}$ , and uses Eq. 4 to compute  $\mathbf{A}_m$ .
- iii) Graph Attention Regression (GAR): It learns the matrix  $\mathbf{A}_m$  of each layer with self-attention during the training process. Formally, let  $\mathbf{h}^{(i), m-1} \in \mathbb{R}^d$  denote the  $i$ -th row of  $\mathbf{H}_{m-1}$ , and the self-attention operation on graph can be written as:

$$\alpha_{ij,m} = \mathbf{w}_m^\top \tanh\left([\mathbf{W}_m^\top \mathbf{h}^{(i), m-1}; \mathbf{W}_m^\top \mathbf{h}^{(j), m-1}]\right) \\ [\mathbf{A}_m]_{ij} = \frac{\exp(\text{LeakyReLU}(\alpha_{ij,m}))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\alpha_{ik,m}))} \quad (5)$$

where  $k \in \mathcal{N}_i$  means that the text  $X^{(k)}$  is the first-order neighbor of  $X^{(i)}$  on the graph  $\mathcal{G}$ , and  $[\cdot; \cdot]$  is vector concatenation operator.

In GAR, the aggregation strength matrix  $\mathbf{A}_m$  is learned from data with an attention layer parameterized by  $\mathbf{w}_m$ , while in GCR-U and GCR-W, it is the prior knowledge provided by topological structure.

We also add a highway connection [40] between every two adjacent GNN layers to control the information flow:

$$\mathbf{T}_m = \sigma(\mathbf{H}_{m-1} \mathbf{W}_{g,m} + \mathbf{b}_{g,m}) \\ \mathbf{H}_m \leftarrow \mathbf{T}_m \odot \mathbf{H}_m + (1 - \mathbf{T}_m) \odot \mathbf{H}_{m-1} \quad (6)$$

where  $\mathbf{T}_m$  is the transform gate with learnable  $\mathbf{W}_{g,m}$  and  $\mathbf{b}_{g,m}$ , and  $\sigma(\cdot)$  denotes logistic function.

A regression layer takes the last GNN layer's output  $\mathbf{H}_M = (\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \dots, \mathbf{e}^{(N)})^\top \in \mathbb{R}^{N \times d}$  as input to obtain the salience score  $\hat{s}^{(i)}$  for the text  $X^{(i)}$ :

$$\hat{s}^{(i)} = \sigma(\mathbf{w}_s^\top \mathbf{e}^{(i)} + \mathbf{b}_s) \quad (7)$$

where the text  $X^{(i)}$ 's representation  $\mathbf{e}^{(i)} \in \mathbb{R}^d$  is learned by GNN. Both  $\mathbf{w}_s$  and  $\mathbf{b}_s$  are parameters to be learned.

3) *Optimization*: For each text  $X^{(i)}$  in  $\mathcal{X}$ , we obtain its gold salience score  $s^{(i)}$  by computing the ROUGE-2 Recall [41] between  $X^{(i)}$  and the opinion summary  $Y$  of  $\mathcal{X}$ , and all scores are normalized to  $[0, 1]$ . Owczarzak *et al.* [42] observed that ROUGE-2 Recall has the highest agreement with human judgments in ROUGE Recall variants, and thus we choose it as gold salience metric. We use mean squared error as the loss function to optimize our salience regression models:

$$\min_{\Phi} \sum_{i=1}^N \left(s^{(i)} - \hat{s}^{(i)}\right)^2 \quad (8)$$

where  $\Phi$  denotes the parameter set of the GNR model (GCR-U, GCR-W, or GAR).

At test time, given a set of opinionated texts, we use the trained salience regression model to predict their salience scores, and rank these texts in descending order. We then select the first  $K$  texts as input to be fed into the second stage for generating a summary. At training time, we use ground-truth salience scores to select.

## B. Opinion Summary Generation With MMR-Guided Graph-to-Sequence Learning

At the second stage, we aim to generate opinion summaries by a maximal marginal relevance guided graph-to-sequence model, named MG2S, which extends the encoder-decoder architecture to estimate  $p(Y | \mathcal{X}^*)$ , where  $\mathcal{X}^*$  contains the top- $K$  texts selected from  $\mathcal{X}$  by the first stage's graph neural regression model.

1) *Graph Attention Encoder*: We represent the top- $K$  texts of  $\mathcal{X}^*$  as a fully-connected graph, where each vertex is a text. We then use a graph attention network, which is similar to the architecture of GAR in Section IV-A, as the encoder to learn the representations of the texts (vertices). Formally, we still use the denotation  $X^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_{l_i}^{(i)})$  to represent the  $i$ -th text in  $\mathcal{X}^*$  without confusion, where  $i \in \{1, \dots, K\}$ .

The graph attention encoder consists of a text encoding module and a stack of graph attention layers. There are two options for text encoding module: GRU-based encoding and BERT-based encoding [43]. For each text  $X^{(i)}$ , GRU-based encoding module produces the hidden state sequence  $(\mathbf{h}_1^{(i)}, \mathbf{h}_2^{(i)}, \dots, \mathbf{h}_{l_i}^{(i)})$  (Eq. 2), and we take the last step hidden state as the input representation of the text for graph attention



layers. BERT-based encoding module employs a deep Transformer encoder [44] pretrained on large-scale external corpora with masked language modeling, capturing linguistic knowledge and producing contextualized representations. It takes  $([\text{CLS}], x_1^{(i)}, x_2^{(i)}, \dots, x_{l_i}^{(i)}, [\text{SEP}])$  as the input sequence of  $X^{(i)}$ , where  $[\text{CLS}]$  and  $[\text{SEP}]$  are special tokens. We use the last layer's output vector sequence  $(\mathbf{h}_1^{(i)}, \mathbf{h}_2^{(i)}, \dots, \mathbf{h}_{l_i}^{(i)})$  to represent the tokens in  $X^{(i)}$ , and we take the corresponding  $[\text{CLS}]$  token's output vector as the input representation of the text for graph attention layers.

After representation learning with graph attention layers (see Eq. 3 and Eq. 5 for details), we use  $\mathbf{e}^{(i)}$  to denote the output representation of the text  $X^{(i)}$ . Thus we have all  $K$  texts' output representations  $\{\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \dots, \mathbf{e}^{(K)}\}$ .

To initialize the state of the decoder, we apply mean pooling operation on all texts' representations  $\{\mathbf{e}^{(i)}\}_{i=1}^K$ , and adopt a fully-connected layer parameterized by  $\{\mathbf{W}_s, \mathbf{b}_s\}$  to obtain the initial state  $\mathbf{s}_0$ :

$$\mathbf{s}_0 = \tanh \left( \mathbf{W}_s \left( \frac{1}{K} \sum_{i=1}^K \mathbf{e}^{(i)} \right) + \mathbf{b}_s \right). \quad (9)$$

**2) Maximal Marginal Relevance-Guided Decoder:** We first describe the classic GRU decoder equipped with copy mechanism for summary generation, and then detail how to incorporate maximal marginal relevance (MMR) mechanism that gives consideration to both salience and non-redundance for summarizing opinionated texts.

Based on the graph attention encoder, recall that each text  $X^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_{l_i}^{(i)})$  has a word-level representation sequence  $(\mathbf{h}_1^{(i)}, \mathbf{h}_2^{(i)}, \dots, \mathbf{h}_{l_i}^{(i)})$  produced by the GRU layer, and a text-level representation  $\mathbf{e}^{(i)}$  produced by the last graph attention layer, where  $i \in \{1, \dots, K\}$ . All  $K$  texts' word-level representation sequence can be written as  $\{\mathbf{h}_1^{(1)}, \mathbf{h}_2^{(1)}, \dots, \mathbf{h}_{l_1}^{(1)}; \mathbf{h}_1^{(2)}, \mathbf{h}_2^{(2)}, \dots, \mathbf{h}_{l_2}^{(2)}; \dots; \mathbf{h}_1^{(K)}, \mathbf{h}_2^{(K)}, \dots, \mathbf{h}_{l_K}^{(K)}\}$ . For the convenience of description, we omit superscripts and re-form subscripts, then re-write it as  $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{l_1}; \mathbf{h}_{l_1+1}, \mathbf{h}_{l_1+2}, \dots, \mathbf{h}_{l_1+l_2}; \dots; \mathbf{h}_{l_1+\dots+l_{K-1}+1}, \mathbf{h}_{l_1+l_2+\dots+l_{K-1}+2}, \dots, \mathbf{h}_{l_1+l_2+\dots+l_K}\}$ . Thus, it can be simply denoted as  $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_L\}$ , where  $L = l_1 + l_2 + \dots + l_K$  is the sum of all texts' lengths.

Consider a classic GRU decoder. At each decoding step  $t$ , the GRU first updates its hidden state from  $\mathbf{s}_{t-1}$  to  $\mathbf{s}_t$  by receiving the previous step's output word embedding  $\mathbf{y}_{t-1}$ , then attends encoder side's word-level representation sequence  $\{\mathbf{h}_j\}_{j=1}^L$  to compute a context vector  $\mathbf{c}_t$  that aggregates information from the source side using attention mechanism [8]:

$$\mathbf{s}_t = \text{GRU}_{dec}([\mathbf{y}_{t-1}; \mathbf{c}_{t-1}], \mathbf{s}_{t-1})$$

$$\alpha_{tj} = \text{softmax}(\mathbf{w}_a^\top \tanh(\mathbf{W}_a[\mathbf{s}_t; \mathbf{h}_j] + \mathbf{b}_a)), \quad 1 \leq j \leq L$$

$$\mathbf{c}_t = \sum_{j=1}^L \alpha_{tj} \mathbf{h}_j \quad (10)$$

where  $\alpha_{tj}$  denotes the attention score of  $\mathbf{h}_j$  at decoding step  $t$ . Input layer's embeddings of the encoder are shared with the decoder's embeddings  $\mathbf{y}_*$ . It then predicts the vocabulary

distribution  $P_{\text{vocab}}^t$  by hidden state and context vector:

$$P_{\text{vocab}}^t = \text{softmax}(\mathbf{W}_v[\mathbf{s}_t; \mathbf{c}_t] + \mathbf{b}_v) \in \mathbb{R}^{|\mathbb{V}|} \quad (11)$$

where  $|\mathbb{V}|$  is the vocabulary size, with learnable parameters  $\mathbf{W}_*$ ,  $\mathbf{w}_*$  and  $\mathbf{b}_*$ . Further, we also employ copy mechanism [45], [46] to allow both copying words from the encoder side and generating novel words from the vocabulary, alleviating the out-of-vocabulary (OOV) problem. Formally, a generation probability  $p_{\text{gen}}^t$  takes hidden state, context vector and previous step's output into account and is computed using a fully-connected layer parameterized by  $\{\mathbf{w}_g, \mathbf{b}_g\}$ , and the final probability of generating word  $w$  is the combination of generation and copy:

$$p_{\text{gen}}^t = \sigma(\mathbf{w}_g^\top [\mathbf{s}_t; \mathbf{c}_t; \mathbf{y}_{t-1}] + \mathbf{b}_g)$$

$$P^t(w) = p_{\text{gen}}^t P_{\text{vocab}}^t(w) + (1 - p_{\text{gen}}^t) \sum_{i: w_i=w} \alpha_{ti}. \quad (12)$$

As we have mentioned earlier, informative opinions to be summarized are usually scattered over multiple opinionated texts, and thus the decoder for generation should possess the ability of paying attention to diverse opinionated texts and avoiding needless redundancy. A core intuition is that if an opinionated text  $X^{(i)}$  has a high salience score, the decoder should up-weight its opinions; however, if the opinions expressed in  $X^{(i)}$  have been considered and generated at previous decoding steps, its content should be down-weighted to avoid generating repetitive opinions.

Based on such consideration, we integrate maximal marginal relevance (MMR) [13] mechanism into the decoder to guide the summary generation process of the graph-to-sequence model, with the aim of avoiding attending only on partial opinions. Formally, for each opinionated text  $X^{(i)}$  in  $\mathcal{X}^*$ , our MMR-guided decoder computes a corresponding marginal relevance score  $\text{mr}_t^{(i)}$ , which is equal to its salience score  $s^{(i)}$  minus its redundancy score  $r_t^{(i)}$  with a factor  $\lambda$ :

$$\text{mr}_t^{(i)} = \lambda s^{(i)} - (1 - \lambda) r_t^{(i)}, \quad i \in \{1, \dots, K\} \quad (13)$$

where  $s^{(i)}$  is obtained by the salience estimation model as described in Section IV-A, and the redundancy score  $r_t^{(i)}$  is estimated by a neural bi-linear function that takes the decoder state  $\mathbf{s}_t$  and the text representation  $\mathbf{e}^{(i)}$  produced by the graph encoder into account:

$$r_t^{(i)} = \sigma(\mathbf{s}_t^\top \mathbf{W}_r \mathbf{e}^{(i)}) \quad (14)$$

where  $\mathbf{W}_r$  is a learnable parameter matrix. All marginal relevance scores are normalized.

We then integrate MMR mechanism into the decoder by refining the word-level attention scores  $\{\alpha_{tj}\}_{j=1}^L$  in Eq. 10 using text-level marginal relevance scores. Specifically, suppose the attention score  $\alpha_{tj}$  is corresponding to the word  $w_j$  in the  $k$ -th text  $X^{(k)}$ , and then it is refined by multiplying the text's marginal relevance score and re-normalization:

$$\tilde{\alpha}_{tj} = \text{softmax} \left( \frac{\alpha_{tj} \cdot \text{mr}_t^{(k)}}{T} \right), \quad 1 \leq j \leq L \quad (15)$$

TABLE I  
STATISTICS OF THE TWO DATASETS EACH SAMPLE IS A PAIR OF {INPUT  
TEXTS, SUMMARY}

Dataset	<i>RottenTomatoes</i>	<i>Idebate</i>
# Samples	3,731	2,259
# Input texts	372,068	17,359
Average of # texts per sample	99.7	7.7
Median of # texts per sample	86.5	7.0
Average length of input texts	24.33	25.47
Average length of summaries	24.46	11.88

where  $T \in (0, 1]$  is a temperature parameter. Based on the MMR-guided decoder, the generation phrase refines the attention scores of the source side according to previous generated content, and considers salient opinions as well as reducing repetitive opinions.

3) *Optimization*: The conditional probability  $p(Y | \mathcal{X}^*)$  can be expanded using chain rule:

$$p(y_1, \dots, y_{|Y|} | \mathcal{X}^*) = \prod_{t=1}^{|Y|} p(y_t | y_1, \dots, y_{t-1}, \mathcal{X}^*) \quad (16)$$

where  $p(y_t | y_1, \dots, y_{t-1}, \mathcal{X}^*)$  is computed by Eq. 12. Our graph-to-sequence model can be optimized by maximizing the conditional log-likelihood:

$$\max_{\Theta} \log p(Y | \mathcal{X}^*; \Theta) \quad (17)$$

where  $\Theta$  denotes the parameter set of the model. At test time, we use beam search to generate opinion summaries.

## V. EXPERIMENTS

### A. Datasets

We use two benchmark datasets released by [2] for evaluation. The first dataset is *RottenTomatoes* collected from the movie review website [www.rottentomatoes.com](http://www.rottentomatoes.com), and Fig. 1(a) shows an example. This dataset contains reviews for 3731 movies, and each sample has a large amount of reviews for a movie and a reference summary written by an editor. The second dataset is *Idebate* collected from [idebate.org](http://idebate.org), a Wikipedia-style website gathering pro and con arguments on controversial issues, and an example is shown in Fig. 1(b). This dataset contains 2259 issues, and each sample has a set of arguments and a one-sentence central claim as summary constructed by an editor. We use the same training/validation/test split as in previous work [2], [7].

Table I lists the statistics of the two datasets. Fig. 3 shows the  $N$ -th text's average gold salience score of all samples.

Evaluation is conducted for two stages: the first stage evaluates the performance of salience estimation, and the second stage for opinionated text summarization.

### B. Experiment I: Salience Estimation

1) *Evaluation Metrics*: For this sub-task, because it is a regression task, we use Pearson Correlation coefficient  $r$ , Spearman's Rank Correlation coefficient  $\rho$  and Kendall Rank Correlation coefficient  $\tau$  as metrics.

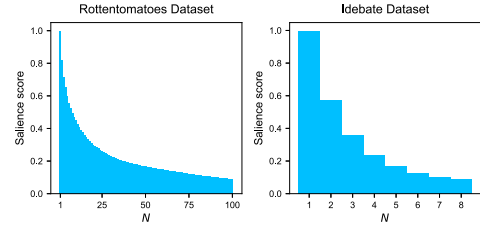


Fig. 3. The  $N$ -th text's average gold salience score of all samples for each dataset.

2) *Comparative Methods*: We compare the following methods for salience estimation:

- Support Vector Regression (SVR) is a basic regression model based on RBF kernel.
- Ridge Regression (RR) is a baseline regression model.
- Neural Network Regression (NNR) computes salience score for each input text with GRU independently, which does not consider the relationships among texts.
- NNR with  $k$ -Nearest Neighbors ( $k$ NNR) enhances NNR by adding the  $k$  most similar texts' representations as context to improve text representations.
- Graph Convolution Regression on Unweighted Graph (GCR-U) and Graph Convolution Regression on Weighted Graph (GCR-W) are our proposed models that consider the relationships among input texts.
- Graph Attention Regression (GAR) is our proposed model. Unlike GCR-U and GCR-W, it learns the aggregation matrix of each layer with self-attention.

3) *Implementation Details*: For our salience regression models GNR (GCR-U, GCR-W and GAR), we use 300 d GloVe embeddings [47] and set the hidden size of GRU to 150. We stack three GNN layers (150 d each). The optimizer is Adam with 0.001 learning rate and 16 batch size.

4) *Results and Discussions*: Table II shows the results of different methods for salience estimation. We can see that all neural models achieve higher metrics than SVR and RR.

Compared to NNR and  $k$ NNR that does not explicitly model the interaction among input texts, three proposed GNN-based regression models consistently perform better than them on both datasets, which verifies the effectiveness of modeling the graph structure constructed from the input opinionated texts. Comparisons among three GNR models reveal the following two interesting points:

- On the *RottenTomatoes* dataset, Convolution Regression on Weighted Graph (GCR-W) outperforms the other two models. However, on the *Idebate* dataset, it performs worst among them.
- On the *Idebate* dataset, the Graph Attention Regression (GAR) obtains the highest results.

We suggest that the above observations are caused by different characteristics of graph topological structures constructed from the two datasets. Consider the density of an undirected graph  $\langle \mathcal{V}, \mathcal{E} \rangle$  as  $\frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}|-1)}$ .<sup>1</sup> In the *RottenTomatoes* dataset, each graph contains 99.7 vertices with 0.77 density on average, compared to 7.7 vertices with 0.91 density in the *Idebate* dataset. This

<sup>1</sup>[https://en.wikipedia.org/wiki/Dense\\_graph](https://en.wikipedia.org/wiki/Dense_graph)

TABLE II  
SALIENCE ESTIMATION RESULTS ON THE TWO DATASETS. “†” AND “‡” INDICATE THAT THE RESULT OUTPERFORMS THE BEST BASELINE WITH  $p < 0.05$  AND  $p < 0.001$  IN THE SIGNIFICANCE TEST (PAIRED  $t$ -TEST), RESPECTIVELY

Model	RottenTomatoes			Idebate		
	Pearson's $r$	Spearman's $\rho$	Kendall's $\tau$	Pearson's $r$	Spearman's $\rho$	Kendall's $\tau$
SVR	0.4416	0.5169	0.3635	0.3668	0.4056	0.2907
RR	0.5060	0.5734	0.4067	0.3809	0.4232	0.3031
NNR	0.5678	0.6393	0.4605	0.4098	0.4544	0.3260
$k$ NNR	0.5706	0.6398	0.4608	0.4135	0.4587	0.3299
GCR-U	0.5826 <sup>†</sup>	0.6443 <sup>†</sup>	0.4645 <sup>†</sup>	0.4307 <sup>†</sup>	0.4746 <sup>†</sup>	0.3394
GCR-W	<b>0.5902<sup>†</sup></b>	<b>0.6491<sup>†</sup></b>	<b>0.4692<sup>†</sup></b>	0.4185	0.4647	0.3335
GAR	0.5830 <sup>†</sup>	0.6450 <sup>†</sup>	0.4657 <sup>†</sup>	<b>0.4315<sup>†</sup></b>	<b>0.4792<sup>†</sup></b>	<b>0.3444<sup>‡</sup></b>

indicates that the graphs from the *RottenTomatoes* dataset tend to be sparse, and thus a weighted graph's adjacency matrix can provide meaningful prior knowledge which notices GNN layers the aggregation strengths among vertices. In contrast, the graphs from the *Idebate* dataset are very dense and many of them are even fully-connected graphs, which results in that adjacency matrices can not give sufficient prior knowledge. Therefore, graph attention operation produces effective vertex features based on its powerful representation learning ability.

This is also the reason why we employ graph attention rather than graph convolution as the encoder of our graph-to-sequence model MG2S. After the salience estimation stage, a graph constructed from top- $K$  salient texts tends to be dense, and thus we leverage self-attention to learn aggregation strengths among vertices on the graph.

### C. Experiment II: Opinionated Text Summarization

1) *Evaluation Metrics*: As in previous work, we use the  $F$ -scores of ROUGE-1, -2, -L and -SU4 [41] and BLEU-4 score [48] for automatic evaluation. ROUGE and BLEU scores are computed by `pyrouge` and `NLTK` tools, respectively.<sup>2</sup>

2) *Comparative Methods*: We conduct comparison that contains two types of summarization methods.

The first type includes extractive-based methods:

- LONGEST returns the longest text from the input set.
- LEXRANK [10] is a representative graph-based method based on PageRank centrality.
- SUBMODULAR [11] is an SVM-based method which optimizes submodular functions.

The second type is abstractive-based methods:

- OPINOSIS [9] is a graph-based method that removes redundant information and merges opinion expressions with syntactic structures.
- RR-S2S [2] first utilizes ridge regression model to estimate saliences, and employs an attention-based sequence-to-sequence model to generate summaries. We add copy mechanism to its decoder for fair comparison.
- AE-SF2S [7] is the state-of-the-art method for opinionated text summarization. It first uses an auto-encoder (AE) to

learn texts' representations, then fuses them via attention and generates summaries using an RNN decoder with copy mechanism. Further, it enhances the decoder by adding salient texts' representations, where the salient texts are extracted using a centroid-based model via BERT.

- HT (Hierarchical Transformer) [49] is the state-of-the-art method for abstractive multi-document summarization. It enhances the encoder of Transformer [44]. Each layer takes one paragraph as input, and the dependency of multiple ones is modeled by inter-paragraph attention.
- BART\_FT is a method by fine-tuning a pretrained BART-Base model [50] on the opinionated text summarization dataset. BART is a representative pretraining model, where its architecture is a Transformer-style encoder-decoder. The main objective of pretraining on external unlabeled corpus is to reconstruct the original text given its corrupted version, which forces the model to understand more about the text's content. Lewis *et al.* [50] have shown the effectiveness of BART when it is fine-tuned for summarization and translation.
- GNR-MG2S and GNR-PMG2S are our proposed methods, where the former adopts GRU encoding and the latter adopts pretrained BERT encoding. Based on the results of salience estimation, we use GCR-W and GAR for *RottenTomatoes* and *Idebate* datasets, respectively.

We further conduct ablation study to verify the effect of each module in our proposed framework:

- GNR-G2S removes the MMR guidance.
- GNR-S2S removes the graph encoder.
- $k$ NNR-S2S uses  $k$ NNR for salience regression.

3) *Implementation Details*: For our summary generation model MG2S with GRU encoding module, we use 100 d GloVe embeddings, and both the GRU's size and the GNN's output size of the graph encoder are set to 100. We use AdaGrad [51] optimizer to train the model, where the learning rate is 0.15 and the batch size is 8. The parameters  $\lambda$  and  $T$  are set to 0.5 and 0.001 respectively. The beam size is set to 5. Note that RR-S2S, AE-SF2S and HT also utilize GloVe embeddings, thus the comparison between GNR-MG2S and them is fair. For PMG2S, we adopt the weights of BERT-Base, Uncased, and use a linear warmup with decay learning rate scheduler. The number of selected top texts is set to  $K = 5$  as in [2], [7].

<sup>2</sup><https://github.com/bheinzerling/pyrouge> <https://www.nltk.org>

TABLE III  
RESULTS OF OPINIONATED TEXT SUMMARIZATION. “RG” AND “B” DENOTE “ROUGE” AND “BLEU” RESPECTIVELY

Method	RottenTomatoes					Idebate				
	RG-1	RG-2	RG-L	RG-SU4	B-4	RG-1	RG-2	RG-L	RG-SU4	B-4
<i>Extractive-based Methods</i>										
LONGEST	15.86	2.07	11.14	2.54	21.33	14.91	2.72	11.64	2.39	14.18
LEXRANK [10]	20.52	5.57	14.29	3.16	17.14	20.10	<b>7.17</b>	16.18	3.46	14.67
SUBMODULAR [11]	20.26	5.50	13.12	2.81	15.14	16.71	5.63	13.04	2.16	11.46
<i>Abstractive-based Methods</i>										
OPINOSIS [9]	17.66	3.34	13.30	3.53	23.74	—	—	—	—	—
RR-S2S [2]	23.65	6.56	18.01	5.96	27.84	18.99	3.36	16.59	5.17	19.89
AE-SF2S [7]	22.49	7.65	18.47	<b>7.79</b>	—	—	—	—	—	—
HT [49]	24.27	7.54	18.23	6.23	29.24	18.25	3.01	15.15	3.89	18.35
GNR-MG2S	<b>25.26</b>	<b>7.87</b>	<b>19.65</b>	6.89	<b>29.63</b>	<b>20.40</b>	3.84	<b>17.94</b>	<b>5.78</b>	<b>21.24</b>
GNR-G2S (w/o MMR)	24.80	7.65	19.33	6.73	29.48	19.95	3.90	17.58	5.74	20.79
GNR-S2S (w/o graph enc.)	24.52	7.29	19.08	6.54	28.54	19.79	3.50	17.00	5.47	20.97
kNNR-S2S (w/o GNR, graph enc.)	24.13	7.22	18.54	6.28	27.81	18.87	3.62	16.67	5.30	20.20
<i>Abstractive-based Methods with Pretrained Weights</i>										
BART_FT [50]	23.85	8.31	18.97	6.27	24.45	19.97	6.03	17.09	5.34	22.22
GNR-PMG2S	23.94	8.62	19.04	6.21	24.09	20.23	5.46	17.92	6.50	23.11

4) *Results and Discussions*: Table III shows the results of different methods for opinion summary generation.<sup>3</sup> On both datasets, our framework GNR-MG2S outperforms other comparative methods on most of evaluation metrics. Comparisons among extractive- and abstractive-based methods reveal that the latter ones obtain higher performance in general. Extractive summarizers like LEXRANK can achieve reasonable performance, however, their output summaries are very long, which can not meet the need of generating concise summaries in opinionated text summarization task. By comparing abstractive-based methods, OPINOSIS performs relatively low due to its unsupervised nature. Next we give detailed discussions.

First, we observe the effectiveness of GNN salience regression models to facilitate the subsequent summary generation stage. GNR-S2S performs better than RR-S2S and kNNR-S2S on both datasets, demonstrating that exploiting the relationships among input opinionated texts with GNN can effectively improve the quality of salience estimation. Compared with AE-SF2S, GNR-S2S obtains comparable performance, which further verifies that the GNN models can select sufficiently salient texts from all texts because a classic encoder-decoder network as the second stage’s model has achieved nearly state-of-the-art performance by taking them as input for generating opinion summaries.

Second, according to the ablation results, we see that our graph-to-sequence model for summary generation achieves better performance. Compared with GNR-S2S, both GNR-G2S and GNR-MG2S can provide a performance boost, and incorporating MMR mechanism into the decoder side further improves the results on most of metrics. This result indicates that our MG2S model benefits from the graph attention encoder as well as the MMR-guided decoder, which enables our framework to outperform previous state-of-the-art methods.

<sup>3</sup>OPINOSIS needs large amounts of redundant opinions to generate a summary, which is not suitable for the *Idebate* dataset. Amplayo and Lapata [7] did not show the results of AE-SF2S on the *Idebate* dataset because this work focuses on generating aspect-controlled review summaries.

Finally, we find that equipping pretrained weights can significantly improve the performance of summarizing opinionated texts on *Idebate* dataset whose training data size is relatively small, and also achieve much higher ROUGE-2 scores on *RottenTomatoes* dataset.<sup>4</sup> The results demonstrate the effectiveness of using pretrained weights that can obtain more concise opinion summaries. BART\_FT performs much better than HT which is a Transformer-based model that trained from scratch on the opinionated text summarization dataset. Our framework GNR-PMG2S that equips pretrained BERT encoding module also outperforms the GNR-MG2S with GRU module when there is lack of training data. Therefore, the opinion summary generation models are benefit from the knowledge stored in pretrained weights.

5) *Human Evaluation*: We further conduct human evaluation to compare the quality of opinion summaries generated by different methods. We invited three human judges to score 40 randomly selected test samples of each dataset.

For each generated summary, we consider four metrics to measure its quality: i) informativeness, which indicates how much salient information is summarized; ii) grammaticality, which indicates whether the content is grammatical; iii) compactness, which indicates whether the content contains unnecessary information; iv) coherence, which indicates whether the summary content is coherent. Incoherent content in a summary will hurt its overall readability. These metrics are rated from 1 to 5 (5 is the best). For each test sample, we also asked our judges to rank the output summaries of different methods as well as the ground-truth summary based on overall quality: we consider three methods, therefore there are four summaries to be ranked for one sample.

Table IV shows the results of human evaluation, and our framework performs better on the metrics of informativeness,

<sup>4</sup>Note that by comparing the averaged length of generated texts between models equipped with and without pretrained weights on *RottenTomatoes* dataset, we observe that models with pretrained weights usually generate shorter texts (17.4 tokens vs. 27.6 tokens), thus the phenomenon that some metrics do not achieve improvements on this dataset is inline with expectations.



TABLE IV  
HUMAN EVALUATION ON GENERATED SUMMARIES AND GROUND-TRUTH. “INFO.,” “GRAM.,” “COMP.,” “COHER.” AND “AVG. RANK” DENOTE INFORMATIVENESS, GRAMMATICALITY, COMPACTNESS, COHERENCE AND AVERAGE RANK, RESPECTIVELY

Method	RottenTomatoes					Idebate				
	Info.	Gram.	Comp.	Coher.	Avg. Rank	Info.	Gram.	Comp.	Coher.	Avg. Rank
LEXRANK	3.5	4.6	2.9	2.2	5.6	3.6	4.8	2.8	2.9	5.2
RR S2S	3.3	3.7	4.2	4.4	4.5	3.3	3.2	4.6	4.4	5.2
GNR MG2S	3.9	3.7	4.5	4.2	4.2	3.6	3.6	4.7	4.5	4.5
BART_FT	3.8	4.1	4.6	4.5	2.6	3.9	4.1	4.6	4.6	2.5
GNR PMG2S	3.9	4.1	4.5	4.6	2.4	3.8	4.0	4.6	4.8	2.4
GROUND TRUTH	4.3	4.8	4.9	4.9	1.5	4.8	4.8	4.9	4.9	1.1

Movie: <i>Breach</i>	
Review 1:	One of the best elements of this lean, <i>tense</i> and <i>coolly believable</i> story about the internal hunt for <i>FBI turncoat</i> Robert Hanssen is how its visual style and dearth of formulaic structural ingredients run counter to almost any other <i>spy movie</i> .
Review 2:	With remarkably simple and <i>effective performances by Ryan Phillippe</i> as O’Neill and <i>Chris Cooper</i> as Hanssen, <i>Breach</i> is a <i>slow, thoughtful</i> duet for two very different, but equally impressive men, both complex and ‘real.’
Review 3:	<i>Slow-burning</i> and <i>sombre</i> , <i>Breach</i> won’t be for everyone. But it’s worth catching for <i>Cooper’s performance</i> , while its fact-based <i>tale of treachery</i> will strike a chord in today’s climate of suspicion.
Review 4:	In truth, the movie leaves us scratching our heads. And yet, for most of it, I was held — by Chris Cooper’s <i>dour portrayal</i> of walled-off ddemons, by the director’s fascination with a deception that, on the surface of it, doesn’t add up.
Review 5:	A quietly fascinating and <i>intelligent</i> story about <i>betrayal</i> .
...	
Ground Truth:	Powered by <i>Chris cooper’s masterful performance</i> , <i>Breach</i> is a <i>tense</i> and <i>engaging portrayal</i> of the <i>FBI’s infamous turncoat</i> .
LexRank:	In truth, the movie leaves us scratching our heads. And yet, for most of it, I was held — by Chris Cooper’s <i>dour portrayal</i> of walled-off ddemons, by the director’s fascination with a deception that, on the surface of it, doesn’t add up.
RR-S2S:	<i>Breach</i> is a <i>tale of</i> the events and <i>betrayal</i> , and features a <i>strong performance from chris cooper</i> , <i>Breach</i> is an <i>intelligent</i> account of the pack.
GNR-MG2S:	<i>Breach</i> is a <i>slow, thoughtful</i> , and <i>believable</i> story about the <i>spy movie</i> that ’s one of fbi ’s <i>finest performances by chris cooper</i> .
BART_FT:	<i>Breach</i> is a <i>well-acted</i> , fact-based thriller that deals with <i>deception and betrayal</i> with <i>intelligence</i> and flair.
GNR-PMG2S:	<i>Breach</i> is a <i>slow - burning, sombre</i> thriller with a <i>strong performance from chris cooper and ryan philippe</i> .

Fig. 4. Opinion summaries generated by different methods.

compactness and average rank. GNR salience model enables our method to obtain higher informativeness, and MMR guidance can avoid generating unnecessary information, which improves informativeness and also contributes to compactness. We can see that abstractive-based methods outperform the extractive summarizer LexRank on coherence, which indicates that abstractive summarizers can produce coherent summaries. In general, opinion summaries generated by our framework achieve higher quality than the compared ones.

#### D. Case Study and Error Analysis

Figure 4 illustrates opinion summaries generated by different methods for a test sample from the *RottenTomatoes* dataset. The extractive baseline LEXRANK returns a relatively long summary, which is not very concise. The abstractive method RR-S2S generates an opinion summary that is more informative than the LEXRANK’s one. However, it also contains redundant information. Compared with LEXRANK and RR-S2S, we observe that our proposed GNR-MG2S captures the diverse opinions scattered over different input texts, and generates a more informative and concise opinion summary without useless redundancy. The two methods with pretrained weights, GNR-PMG2S and BART\_FT, produce more concise opinion summaries, and keep the informativeness simultaneously. Therefore, it is desirable that exploiting

pretraining models to enhance the quality of opinion summary generation.

We also illustrate that the opinion summary generated by our framework GNR-MG2S is unfaithful. As error analysis, we can see that abstractive-based methods may generate a summary that contains inconsistent content with input texts. Uncompleted phrases and unfaithful expressions generated by abstractive summarizers hurt the quality of an opinion summary. Recent work has paid attention to the unfaithful issue of summarization models [52], [53], and we leave the improvement of faithfulness in opinionated text summarization to future work.

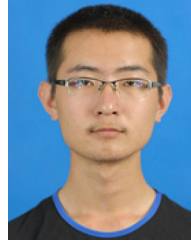
## VI. CONCLUSION

In this paper, we propose a two-stage graph-to-sequence learning framework for opinionated text summarization. To identify important texts from multiple opinionated texts, we model the relationships among them with a graph neural regression model GNR for estimating salience scores. To attend informative opinions that are scattered over different texts and avoid focusing on only partial opinions during summary generation, we develop a graph-to-sequence model MG2S, in which an MMR mechanism guides the decoder to pay attention to diverse opinions. Experimental results and human evaluation verify the effectiveness of our proposed framework.

## REFERENCES

- [1] M. Jang and J. Allan, "Explaining controversy on social media via stance summarization," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 1221–1224.
- [2] L. Wang and W. Ling, "Neural network-based abstract generation for opinions and arguments," in *Proc. Annu. Conf. North Amer. Ch. Assoc. Comput. Linguist.*, 2016, pp. 47–57.
- [3] S. Somasundaran and J. Wiebe, "Recognizing stances in ideological online debates," in *Proc. NAACL-HLT Workshop Comput. Approaches Anal. Gener. Emotion Text*, 2010, pp. 116–124.
- [4] P. Sobhani, D. Inkpen, and S. Matwin, "From argumentation mining to stance classification," in *Proc. 2nd Workshop Argumentation Mining*, 2015, pp. 67–77.
- [5] P. Wei, J. Lin, and W. Mao, "Multi-target stance detection via a dynamic memory-augmented network," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 1229–1232.
- [6] N. Reimers, B. Schiller, T. Beck, J. Daxenberger, C. Stab, and I. Gurevych, "Classification and clustering of arguments with contextualized word embeddings," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 567–578.
- [7] R. K. Amplayo and M. Lapata, "Informative and controllable opinion summarization," in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2021.
- [8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations, (ICLR)*, 2015.
- [9] K. Ganesan, C. Zhai, and J. Han, "Opinosis: A graph based approach to abstractive summarization of highly redundant opinions," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 340–348.
- [10] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, 2004.
- [11] R. Sipos, P. Shivaswamy, and T. Joachims, "Large-margin learning of submodular summarization models," in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2012, pp. 224–233.
- [12] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev, "Graph-based neural multi-document summarization," in *Proc. Conf. Comput. Natural Lang. Learn.*, 2017, pp. 452–462.
- [13] J. G. Carbonell and J. Goldstein, "The use of MMR and diversity-based reranking for reordering documents and producing summaries," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1998, pp. 335–336.
- [14] E. Chu and P. Liu, "MeanSum: A neural model for unsupervised multi-document abstractive summarization," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1223–1232.
- [15] M. Alshomary, S. Syed, M. Potthast, and H. Wachsmuth, "Target inference in argument conclusion generation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4334–4345.
- [16] P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan, "Exploring sentiment summarization," in *Proc. AAAI Spring Symp. Exploring Attitude Affect Text: Theories Appl.*, vol. 39, 2004.
- [17] J. Li, H. Li, and C. Zong, "Towards personalized review summarization via user-aware sequence network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 6690–6697.
- [18] M. Isonuma, J. Mori, and I. Sakata, "Unsupervised neural single-document summarization of reviews via learning latent discourse structure and its ranking," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2142–2152.
- [19] H. P. Chan, W. Chen, and I. King, "A unified dual-view model for review summarization and sentiment classification with inconsistency loss," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1191–1200.
- [20] A. Roush and A. Balaji, "DebateSum: A large-scale argument mining and summarization dataset," in *Proc. 7th Workshop Argument Mining*, 2020, pp. 1–7.
- [21] S. Syed, R. El Baff, J. Kiesel, K. Al Khatib, B. Stein, and M. Potthast, "News editorials: Towards summarizing long argumentative texts," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 5384–5396.
- [22] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2004, pp. 168–177.
- [23] F. Li *et al.*, "Structure-aware review mining and summarization," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 653–661.
- [24] G. Carenini, R. Ng, and A. Pauls, "Multi-document summarization of evaluative text," in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2006, pp. 305–312.
- [25] K. Lerman, S. Blair-Goldensohn, and R. McDonald, "Sentiment summarization: Evaluating and learning user preferences," in *Proc. 12th Conf. Eur. Chapter ACL*, 2009, pp. 514–522.
- [26] X. Zhou, X. Wan, and J. Xiao, "CMiner: Opinion extraction and summarization for chinese microblogs," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1650–1663, Jul. 2016.
- [27] N. Yu, M. Huang, Y. Shi, and X. Zhu, "Product review summarization by exploiting phrase properties," in *Proc. 26th Int. Conf. Comput. Linguistics: Tech. Papers*, 2016, pp. 1113–1124.
- [28] S. Angelidis and M. Lapata, "Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3675–3686.
- [29] G. Di Fabbri, A. Stent, and R. Gaizauskas, "A hybrid approach to multi-document summarization of opinions in reviews," in *Proc. 8th Int. Natural Lang. Gener. Conf.*, 2014, pp. 54–63.
- [30] S. Gerani, Y. Mehdad, G. Carenini, R. T. Ng, and B. Nejat, "Abstractive summarization of product reviews using discourse structure," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1602–1613.
- [31] M. Yang, Q. Qu, Y. Shen, Q. Liu, W. Zhao, and J. Zhu, "Aspect and sentiment aware abstractive review summarization," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1110–1120.
- [32] Y. Suhara, X. Wang, S. Angelidis, and W.-C. Tan, "OpinionDigest: A simple framework for opinion summarization," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5789–5798.
- [33] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, and M. Sun, "A unified model for extractive and abstractive summarization using inconsistency loss," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 132–141.
- [34] S. Gehrmann, Y. Deng, and A. Rush, "Bottom-up abstractive summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4098–4109.
- [35] L. Lebanoff, K. Song, and F. Liu, "Adapting the neural encoder-decoder framework from single to multi-document summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4131–4141.
- [36] A. Fabbri, I. Li, T. She, S. Li, and D. Radev, "Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1074–1084.
- [37] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.
- [38] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations, (ICLR)*, 2017.
- [39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations, (ICLR)*, 2018.
- [40] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2377–2385.
- [41] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, 2004, pp. 74–81.
- [42] K. Owczarzak, J. Conroy, H. T. Dang, and A. Nenkova, "An assessment of the accuracy of automatic evaluation in summarization," in *Proc. Workshop Eval. Metrics Syst. Comparison Autom. Summarization*, 2012, pp. 1–9.
- [43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Annu. Conf. North American Chapter Assoc. Comput. Linguistics*, 2019, pp. 4171–4186.
- [44] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
- [45] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1631–1640.
- [46] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1073–1083.
- [47] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
- [48] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.

- [49] Y. Liu and M. Lapata, "Hierarchical transformers for multi-document summarization," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5070–5081.
- [50] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.
- [51] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, 2011, pp. 2121–2159.
- [52] B. Goodrich, V. Rao, P. J. Liu, and M. Saleh, "Assessing the factual accuracy of generated text," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 166–175.
- [53] W. Kryscinski, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 9332–9346.



**Jiahao Zhao** received the B.S. degree from Southwest University, Chongqing, China. He is currently working toward the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include social media analytics and multimodal data mining.



**Penghui Wei** received the B.S. degree from Wuhan University, Wuhan, China. He is currently working toward the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include information retrieval, opinion mining, and opinionated text generation.



**Wenji Mao** received the Ph.D. degree in computer science from the University of Southern California, Los Angeles, CA, USA. She is currently a Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and with the University of Chinese Academy of Sciences, Beijing, China. Her research interests include artificial intelligence and social computing.