Dialogue Summarization with Static-Dynamic Structure Fusion Graph

Shen Gao[1*], **Xin Cheng**[2*], **Mingzhe Li**[3], **Xiuying Chen**[4],
Jinpeng Li[2], **Dongyan Zhao**[2,5,6†], **Rui Yan** [7,8†]

[1] School of Computer Science and Technology, Shandong University

[2] Wangxuan Institute of Computer Technology, Peking University [3]Ant Group

[4] Computational Bioscience Research Center, KAUST [5] National Key Laboratory of General Artificial Intelligence

[6]BIGAI, Beijing, China [7] Gaoling School of Artificial Intelligence, Renmin University of China

[8] Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education

{shengao,zhaody}@pku.edu.cn, chengxin1998@stu.pku.edu.cn, ruiyan@ruc.edu.cn

Abstract

Dialogue summarization, one of the most challenging and intriguing text summarization tasks, has attracted increasing attention in recent years. Since dialogue possesses dynamic interaction nature and presumably inconsistent information flow scattered across multiple utterances by different interlocutors, many researchers address this task by modeling dialogue with pre-computed static graph structure using external linguistic toolkits. However, such methods heavily depend on the reliability of external tools and the static graph construction is disjoint with the graph representation learning phase, which could not make the graph dynamically adapt to the downstream summarization task. In this paper, we propose a Static-Dynamic graph-based Dialogue Summarization model (SDDS)*, which fuses prior knowledge from human expertise and implicit knowledge from a PLM, and adaptively adjusts the graph weight, and learns the graph structure in an end-to-end learning fashion from the supervision of summarization task. To verify the effectiveness of SDDS, we conduct extensive experiments on three benchmark datasets (SAMSum, MediaSum, and DialogSum) and observe significant improvement over strong baselines.

1 Introduction

Dialogue summarization, aiming at distilling the salient information from a dialogue context into a concise summary, is one of the most challenging and intriguing tasks in text summarization (Gurevych and Strube, 2004; Feng et al., 2021a; Cheng et al., 2023a). It can help people quickly capture the highlights of a semi-structured and multi-participant dialogue without reviewing the complex dialogue context (Feng et al., 2022)

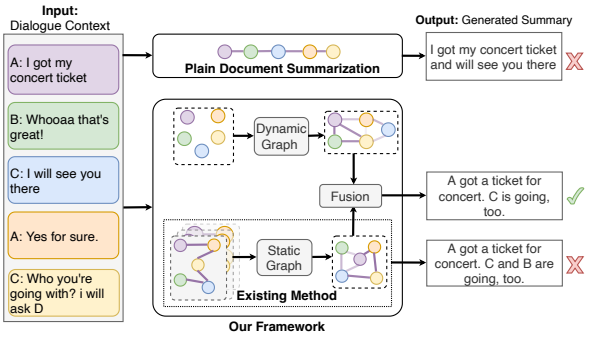

Figure 1: Traditional summarization methods and existing dialogue summarization methods can not understand the dialogue structure comprehensively while our framework could adaptively learn dialogue structure with a dynamic graph module.

and has many real-world applications (Liu et al., 2019; Zhang et al., 2021).

Since dialogue is the most fundamental and specially privileged arena of language (Jurafsky and Martin, 2000), it possesses dynamic interaction nature and presumably inconsistent information flow scattered across multiple utterances by different interlocutors (Li et al., 2022). So the plain document summarization methods (Gehrmann et al., 2018; Zhang et al., 2020a) could not adapt well in this setting. As shown in Figure 1, the plain text summarization method takes dialogue as a long sequence without modeling its structure thus can not generate a proper summary.

To address this problem, the existing dialogue summarization methods mainly focus on modeling dialogue with pre-computed static graph structure using external linguistic toolkits such as discourse parsing (Chen and Yang, 2021; Feng et al., 2021a), dialogue topic modeling (Chen and Yang, 2020; Zhao et al., 2020), dialogue state tracking (Zhao et al., 2021b) and dialogue acts modeling (Goo and Chen, 2018; Chen and Yang, 2021). Although static graph structure captures inconsistent information flow of dialogue to some extent and achieves

*The first two authors contributed equally.

†Corresponding Author.

*Code available at https://github.com/Hannibal046/SDDS

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 13858–13873
July 9-14, 2023

sufficient improvements across various datasets, we argue that there exist two fundamental drawbacks: (1) such methods heavily depend on the reliability of external linguistic tools which may not deliver the accurate output and cause error propagation. For example, the commonly used discourse parser in dialogue summarization (Chen and Yang, 2021; Feng et al., 2021a) is a trained model from Shi and Huang (2019), which is optimized for a dialogue summarization-agnostic online game dialogue dataset. This distribution shift may greatly hurt the generalization ability of the parser (Qian and Yu, 2019). (2) the static graph construction is disjoint with the graph representation learning phase and such a fixed graph could not dynamically adapt to the downstream summarization task.

In this paper, we propose the **S**tatic-**D**ynamic graph-based **D**ialogue **S**ummarization model (SDDS) which contains two graph modules: (1) Static Graph Module and (2) Dynamic Graph Module. For the static graph module, we consider four dialogue structures. Except for the commonly used (1) discourse parsing and (2) keywords co-occurrence relationship, we propose two novel structure modeling methods: (3) speaker relationship and (4) utterance position modeling.

Complementary to these four static graphs that encode human prior into the model, we propose a dynamic graph module that is constructed from a pre-trained language model (PLM). The language model pre-trained on the massive corpora captures oceans of knowledge without human annotation (Warstadt et al., 2019) and shows strong capability in modeling the various textual relationships (Lyu et al., 2021; Chen et al., 2021a). Thus, we propose to use the deep semantic representation for utterances obtained from the PLM to learn the various utterance and speaker relationships. By fusing prior knowledge from human expertise and implicit knowledge from a PLM with a fine-grained 1×1 convolution, SDDS could adaptively adjust the graph weight and learn the graph structure in an end-to-end learning fashion from the supervision of summarization task.

Figure 1 shows the overall architecture of the SDDS model. First, we employ a pre-trained language model to encode all the utterances into vector representations. Next, we construct four static graphs and propose an early fusion method to combine these static graphs. Then, a dynamic graph module is used to learn the semantic relationships using utterance vector representations. Finally, we propose a fusion mechanism to combine the static and dynamic graphs into a unified representation and employ a pre-trained language model to generate the summary by incorporating the updated utterance representation of the combined graph. To verify SDDS, we conduct extensive experiments on three benchmark datasets. Experimental results demonstrate that the SDDS achieves substantial improvement over strong baselines. We also carefully examine each key component and gives a detailed analysis of SDDS for future research.

To sum up, our key contributions are:

• We are the first to take a deep look into the limitation of the current static graph-based methods.

• We propose a novel framework called SDDS which fuses prior knowledge from human expertise and implicit knowledge from a PLM, adaptively adjusts the graph weight, and learns the graph structure in an end-to-end learning fashion from the supervision of summarization task.

• Comprehensive experiments conducted on three benchmark datasets show SDDS achieves significant improvement over strong baselines.

2 Related Work

2.1 Dialogue Summarization

Recent research works in dialogue summarization can be classified into two categories. Since this research task is a newly proposed task, the first category of works focuses on exploring new datasets. AMI (Carletta and et al., 2005) and ICSI (Janin and et al., 2003) corpus are meeting summarization datasets which contain 57 and 137 data samples respectively. To train the neural-based summarization model, researchers also propose several large-scale datasets. SAMSum (Gliwa et al., 2019) is a large-scale chit-chat summarization dataset with 14,732 training samples, and most of the samples are two-party dialog with a 2.2 average speaker. MediaSum (Zhu et al., 2021) is a multi-party dialogue summarization dataset collected from news interviews with $463K$ data samples and 6.5 average speakers.

The second category of research works proposes to incorporate manifold information to help the dialogue summarization. Feng et al. (2021a) and Chen and Yang (2020) propose using a discourse parsing tool or heuristic structure extraction method to help the model capture the dialogue structures. These methods leverage the graph model to capture the di-

alogue structure and they focus on the algorithm of passing messages between utterance nodes in their methods. Feng et al. (2021b) propose to extract the keyword, topic, and redundancy utterances by using DialoGPT and incorporate this manifold information in the summary generation process. Feng et al. (2020) propose using large-scale commonsense knowledge to facilitate dialogue understanding and summary generation. Geng et al. (2022b) propose three speaker-aware supervised contrastive learning tasks to recognize the unique format of the speaker-utterance pair. Ravaut et al. (2022) fuse several summary candidates to produce a novel abstractive second-stage summary. Li et al. (2022) propose a novel curriculum-based prompt learning method with self-training to tackle the insufficient training data problem. Li et al. (2023) propose to learn disentangled representation via domain adaptation for dialogue summarization tasks.

2.2 Graph Neural Network

Graph is widely used in many structure data modeling tasks: recommendation (Liu et al., 2020; Jiang et al., 2018; Fan et al., 2019a), social network modeling (Wu et al., 2019a; Fan et al., 2019b; Yang et al., 2020), and knowledge-graph based tasks (Jiang and Han, 2020; Wu et al., 2019b). In the document summarization task, many existing works (Tan et al., 2017; Wang et al., 2020) employ the graph model to capture the document structures and incorporate this structure into abstractive or extractive summarization. Wei (2012) proposes a heterogeneous graph consisting of topic, word and sentence nodes and uses the markov chain model for the iterative update. Tan et al. (2017) HSG (Wang et al., 2020) employs a heterogeneous graph network to model the words and sentences with in single and multi-documents and then extracts sentences from document. In the dialogue summarization field, using a graph network to modeling the dialogue structure is also a common practice. However, most of the existing works (Feng et al., 2020, 2021a) use the pre-computed graph to capture the dialogue structure and focus on the algorithm of passing messages between utterance nodes in their methods.

3 Problem Formulation

Given a dialogue context $D = \{u_1, \cdots, u_{L_d}\}$ with L_d utterances and each utterance $u_i = \{w_{i,1}, \cdots, w_{i,L_u^i}\}$ contains L_u^i words. We use the s_i to denote the speaker of i-th utterance and $|S|$ to denote the number of speakers. Our goal is to generate the summary $\hat{Y} = \{\hat{y}_1, \cdots, \hat{y}_{L_y}\}$ which has L_y words. And we use the difference between generated summary $\hat{Y}$ and the ground truth Y as the training objective.

4 SDDS Model

In this section, we introduce the **S**tatic-**D**ynamic graph based **D**ialogue **S**ummarization model (SDDS). An overview is shown in Figure 2.

4.1 Utterance Encoder

We employ the pre-trained BART (Lewis et al., 2020) to encode each utterance independently:

$$\{\mathbf{h}_{i,0}, \mathbf{h}_{i,1}, \cdots, \mathbf{h}_{i,L_u^i}\} = \text{Enc}([\text{CLS}], w_{i,1}, \cdots, w_{i,L_u^i}), \tag{1}$$

where $\text{Enc}(\cdot)$ is the encoder module in BART which outputs the vector representation $\mathbf{h}_{i,j}$ of j-th input word $w_{i,j}$ in i-th utterance. To obtain a vector representation of each utterance, we extract the hidden state $\mathbf{h}_{i,0}$ of the input special token [CLS] as the vector representation $\mathbf{u_i} = \mathbf{h}_{i,0}$ of i-th utterance. And $\mathbf{U} = \{\mathbf{u_1}, \ldots, \mathbf{u_{L_d}}\}$ are the representations for all utterances.

4.2 Static Graph Construction

In this section, we first propose 4 heuristic dialogue structure modeling methods to build the relationships between utterances using a graph network.

1. Discourse Parsing Graph. Since dialogue discourse relations can explicitly show the information flow and interaction between utterance (Feng et al., 2021a), we employ a discourse parsing toolkit (Shi and Huang, 2019) to build dependency-based dialogue structure which allows relations between non-adjacent utterances which is applicable for multi-party conversions. There are 16 discourse relations in total: comment, clarification-question, elaboration, acknowledgment, continuation, explanation, conditional, question-answer, alternation, question-elaboration, result, background, narration, correction, parallel, and contrast. After obtaining the discourse parsing result, we use an embedding matrix to project these discreate relations into vector representation:

$$\mathcal{G}_d^s(i,j) = \mathcal{E}_d\left(\text{DiscoParse}(u_i, u_j)\right), \tag{2}$$

where $\mathcal{E}_d \in \mathbb{R}^{16,1}$ denotes the embedding matrix.

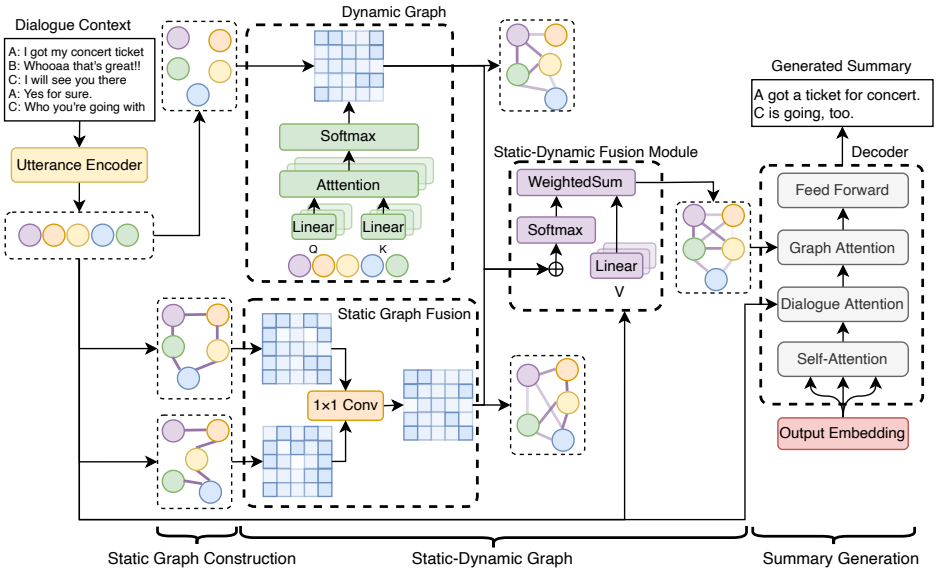

Figure 2: Overview of SDDS.

2. Keywords Co-occurrence Graph. It is intuitive that when two utterances contain the same keyword, they may focus on the same topic and they are semantically correlated. We employ the function KeyCo-occ to denote the function that calculates the number of common keywords in two utterances.Then we use an embedding matrix to project the integer number of keyword co-occurrence to a vector:

$$\mathcal{G}_k^s(i,j) = \mathcal{E}_k\left(\text{KeyCo-occ}(u_i, u_j)\right), \quad (3)$$

where $\mathcal{E}_k \in \mathbb{R}^{N_k,1}$ denotes the embedding matrix, and N_k and D denotes the maximum number of co-occurrence keyword and the hidden size respectively. In this paper, we only use the noun and entity words as the keyword.

3. Speaker Relation Graph. Since it is essential to understand the fine-grained interaction between speakers in dialogue context, in this paper, we propose a simple yet effective speaker relationship modeling method. We use a sliding window around each utterance, and count the frequency of occurrence for each speaker in this sliding window, and the obtain a speaker interaction frequency matrix $\hat{\mathcal{G}}_s^s \in \mathbb{N}^{|S|,|S|}$. Intuitively, if an element in $\hat{\mathcal{G}}_s^s$ achieves the relatively high value in both row-wise and column-wise, that means the speakers of the row and column have a strong relationship compared to other speakers. For example, in Figure 4, we can find that speaker C usually talks after A, which indicates the strong relationship between two speakers. Thus, to normalize the frequency of interaction between speakers, we first apply the row-wise softmax on the interaction frequency matrix $\hat{\mathcal{G}}_s^s$ and then apply column-wise softmax on $\hat{\mathcal{G}}_s^s$ independently. Next, we apply the element-wise product and result in the final speaker relation matrix $\tilde{\mathcal{G}}_s^s \in \mathbb{R}^{|S|,|S|}$:

$$\tilde{\mathcal{G}}_s^s = \text{softmax}_r(\hat{\mathcal{G}}_s^s) \times \text{softmax}_c(\hat{\mathcal{G}}_s^s), \quad (4)$$

where softmax_r and softmax_c denotes the row-wise and column-wise softmax function respectively. For example, when speaker C usually talks after A, which indicates the strong relationship between two speakers, and we can find that the value between speaker A and C achieves the highest value in the $\tilde{\mathcal{G}}_s^s$. Finally, we fill the utterance-level speaker relation adjacent matrix $\mathcal{G}_s^s \in \mathbb{R}^{L_d,L_d}$ using the value in $\tilde{\mathcal{G}}_s^s$:

$$\mathcal{G}_s^s(i,j) = \tilde{\mathcal{G}}_s^s(s_i, s_j), \quad (5)$$

where $\tilde{\mathcal{G}}_s^s(s_i, s_j) \in \mathbb{R}$ denotes the value in s_i-th row and s_j column. More details can be found in the Appendix § A.1.

4. Utterance Position Graph. To capture the position information of utterances, we use the relative distance between utterances as the edge feature of utterance position graph $\mathcal{G}_p^s$. Similarly, we also employ an embedding matrix to map the discrete distance into vector space:

$$\mathcal{G}_p^s(i,j) = \mathcal{E}_p\left(j - i\right), \quad (6)$$

where $\mathcal{G}_p^s$ is the adjacent matrix of utterance position graph and the value denotes the relative distance. And $\mathcal{E}_p \in \mathbb{R}^{L_d,1}$ is the embedding matrix.

4.3 Static-Dynamic Graph Module

4.3.1 Static Graph Fusion

After obtaining adjacent matrixes for static graphs, to conduct cross-graph fusion and interaction, we can see these adjacent matrixes as different channels and use a simple but efficient 1×1 convolutional layer to integrate these adjacent matrixes into a fused relationship representation between utterances:

$$\mathcal{G}^s = \text{Conv}\left(\mathcal{G}_p^s \oplus \mathcal{G}_s^s \oplus \mathcal{G}_k^s \oplus \mathcal{G}_d^s\right), \quad (7)$$

where $\oplus$ denotes the concatenation operator of matrixes and $\mathcal{G}^s \in \mathbb{R}^{L_d, L_d}$ is the fused relationship representation.

4.3.2 Dynamic Graph Module

To capture the semantic relationship between utterances based on their deep vector representation, inspired by the Transformer (Vaswani et al., 2017), we propose a dynamic graph module that does not use any pre-computed or heuristic method to build the connections between nodes. We first project the utterance vector representations $\mathbf{U} = \{\mathbf{u_1}, \ldots, \mathbf{u_{L_d}}\}$ into two different vector spaces, and calculate the relationship as $A \in \mathbb{R}^{L_d, L_d}$:

$$Q = \mathbf{U}W_Q, K = \mathbf{U}W_K, A = \frac{QK^\top}{\sqrt{d_K}}, \quad (8)$$

where W_Q, W_K are all trainable parameters. Next, the relation matrix A can be seen as the adjacent matrix for the utterance graph, and this graph is built dynamically via the multi-head attention mechanism. Since this graph is built by the attention module with trainable parameters, it can capture the task-specific relationship between utterances that may not be covered by the heuristic static graph.

4.3.3 Fusion Module

To integrate the static and dynamic graph, we propose a fusion method to combine the relation matrix A of dynamic graph and the adjacent matrix $\mathcal{G}^s$ of the static graph into a unified graph $\mathcal{G}^u$. Similar with the static graph fusion method, we also employ a 1×1 convolutional layer to combine the two matrixes A and $\mathcal{G}^s$ as two channel:

$$\mathcal{G}^u = \text{Conv}\left(A \oplus \mathcal{G}^s\right). \quad (9)$$

We obtain unified adjacent matrix $\mathcal{G}^u \in \mathbb{R}^{L_d, L_d}$.

To unify the static and dynamic graph structures into a final utterance representation, we employ a self-attention layer as shown in Figure 2. We first project the utterance representation into multiple vector spaces using multi-head attention which is same with Equation 8, and then apply the weighted sum operation using the unified graph $\mathcal{G}^u$ as the attention score:

$$\{\mathbf{g_1}, \ldots, \mathbf{g_{L_d}}\} = \text{softmax}(\mathcal{G}^u)V, \quad (10)$$

$$V = \mathbf{U}W_V \quad (11)$$

where $\mathbf{g_i}$ is the graph representation of the i-th utterance.

4.4 Summary Generator

Finally, to incorporate the graph representation which captures the dialogue structure information in the generation process of the summary, we use dual cross attention (Cheng et al., 2022) mechanism by proposing a graph attention layer on the top of original self attention layer. We first apply the self-attention on the masked output summary embeddings, and then use the output $\mathbf{p}^s$ to cross-attend to the token-level dialogue hidden states $\{\mathbf{h}_{1,1}, \cdots, \mathbf{h}_{L_d, L_u^i}\}$ produced by the utterance encoder (introducued in Equation 1):

$$\mathbf{p}^q = \text{MHAtt}(\mathbf{p}^s, \{\mathbf{h}_{1,1}, \cdots, \mathbf{h}_{L_d, L_u^i}\}), \quad (12)$$

where MHAtt is the standard multi-head attention layer and this procedure is the same as the original BART decoder. After the cross-attention layer, we apply a multi-head graph attention layer which aggregate useful knowledge from the updated graph nodes according to the state of each decoding step:

$$\mathbf{p}^g = \text{MHAtt}(\mathbf{p}^q, \{\mathbf{g_1}, \cdots, \mathbf{g_{L^d}}\}). \quad (13)$$

Finally, we apply a fully connected feed-forward network on $\mathbf{p}^g$ to predict the distribution over the vocabulary of the generated summary. And we use the cross-entropy loss between generated summary and ground truth summary as the training objective to optimize all the parameters of SDDS. We use the parameters in the pre-trained language model BART to initialize the corresponding parameters in our Transformer based text encoder (Equation 1) and summary generator.

5 Experimental Setup

5.1 Dataset and Evaluation

We verify the effectiveness of SDDS on three benchmark datasets: SAMSum (Gliwa et al.,

2019), MediaSum-NPR (Zhu et al., 2021) and DialogSum (Chen et al., 2021b). For evaluation metrics, following standard practice in summarization (Zhang et al., 2020a; Cheng et al., 2023b), we adopt ROUGE (R-1/2/L) (Lin, 2004), BERTScore (Zhang et al., 2020b), BARTScore (Yuan et al., 2021) and MoverScore (Zhao et al., 2019). More implementation details, dataset statistics, and evaluation metrics can be found in Appendix A.2.

5.2 Compared Methods

To verify the effectiveness of SDDS, we compare with the following baselines: **S2SA** is the Sequence-to-Sequence framework equipped with the attention and copy mechanism (See et al., 2017). **Transformer** (Vaswani et al., 2017) is a self-attention-based text generation framework. **BART** (Lewis et al., 2020) and **UniLM** (Bao and et al., 2020) are large-scale pre-trained language models. **MV-BART** (Chen and Yang, 2020) is a BART-based method that incorporates topic and stage information to capture the structure of the dialogue context. **FROST** (Narayan et al., 2021) prompts target summaries with entity chains—ordered sequences of entities mentioned in the summary. **CODS** (Wu et al., 2021) propose a granularity controlled dialogue summarization method. **GPT-Anno** (Feng et al., 2021b) uses the DialoGPT (Zhang and et al., 2020) as an unsupervised dialogue annotator for keyword and topic information. **CONDIGSUM** (Liu et al., 2021a) proposes two topic-aware contrastive learning objectives to implicitly model the topic change and handle information scattering. **SSAnet** (Zhao et al., 2021a) proposes a heterogeneous semantic slot graph to enhance the slot features for more correct summarization. **Coref-Attn** (Liu et al., 2021b) proposes to explicitly incorporate coreference information. **SCL** (Geng et al., 2022a) proposes speaker-aware supervised contrastive learning for better factual consistency. **HITL** (Chen et al., 2022) incorporates human feedback into the training of summarization model. **SummaFusion** (Ravaut et al., 2022) fuses several summary candidates to produce a second-stage summary.

6 Experimental Result

6.1 Overall Performance

Automatic Evaluation We compare our model with the baselines listed in Table 1. Our model performs significantly better than other dialogue summarization models including the state-of-the-art model GPT-Anno with improvements of 1.88%, 2.05%, and 0.98% in terms of R-1, R-2, and R-L on the benchmark dataset SAMSum with $p < 0.05$. We also find that SDDS can achieve consistently better performance than the strong baselines on other two datasets. This demonstrates that the static-dynamic graph model can fuse the human prior knowledge of dialogue structure and learn the semantic relationship dynamically, which helps the summarization model understand the dialogue context better. Although the baseline methods use the heuristic graph construction method (*e.g.,* using discourse parsing result) or use the pre-trained language model GPT-2 to explore the deep semantic

Method	R-1	R-2	R-L
SAMSum			
FROST	51.86	27.67	47.52
SSAnet	51.28	27.15	49.37
CODS	52.65	27.84	50.79
MV-BART	53.42	27.98	49.97
GPT-Anno	53.70	28.79	55.30*
CONDIGSUM	54.30	29.30	45.20
Coref-Attn	53.93	28.58	50.39
SCL	54.22	29.87	51.35
HITL	53.76	28.04	50.56
SummaFusion	52.76	28.24	43.98
BART	52.96	28.62	54.38
SDDS	**54.97**†	**30.01**†	**56.27**†
DialogSum			
Longest-3	24.15	6.25	22.73
TextRank	21.19	6.49	23.91
Transformer	35.91	8.74	33.50
UniLM	47.04	21.13	45.04
GPT-Anno	47.12	20.88	44.56
BART	47.28	21.18	44.83
SDDS	**48.02**†	**21.68**†	**45.88**†
MediaSum-NPR			
Longest-3	28.39	11.21	19.90
S2SA	35.86	16.01	24.46
UniLM	41.42	20.73	30.65
GPT-Anno	41.98	21.42	31.56
BART	43.55	21.99	32.03
SDDS	**43.91**†	**22.53**†	**32.28**†

Table 1: Automatic evaluation results. $*$ denotes our re-evaluated result. † denotes the method is significantly better than baselines with p-value < 0.05, tested by bootstrap re-sampling (Koehn, 2004).

information, their performance is still worse than SDDS which combines the human prior knowledge of dialogue structure and the deep semantic relationship using the static-dynamic graph. Different from the other datasets, MediaSum-NPR has more speakers (avg. 4.0 speakers) and the dialogue structure is more complex. From Table 1, we can find that SDDS achieves better performance. This demonstrates SDDS can be directly generalized to the multi-speaker scenario. We also conduct more fine-grained analysis on SDDS measured by token-level F-measure. As Figure 3 shows, SDDS surpasses baselines in almost all word frequencies and performs especially well for low-frequency words, which shows the great generalization and robustness of SDDS.

Since ROUGE can only evaluate token level syntactical similarity, we also measure the semantic similarity of generated summary and ground truth on SAMSum by BERTScore (Zhang et al., 2020b), BARTScore (Yuan et al., 2021) and MoverScore (Zhao et al., 2019). Results on Table 2 show that these model-based scores are consistent with the ROUGE and human evaluation (detailed below), and verify the superiority of SDDS.

Method	BERTScore	BARTScore	MoverScore
BART	91.67	-1.48	62.27
MV-BART	90.85	-1.86	62.50
GPT-Anno	90.79	-2.19	62.47
SDDS	**92.04**	**-1.37**	**62.98**

Table 2: Semantic similarity evaluation on SAMSum.

Human Evaluation For the human evaluation, we asked three graduate students with professional English proficiency to rate the generated summary according to its *fluency* and *factual coherence* on SAMSum dataset. The rating score ranges from 1 to 3, with 3 being the best. **BART** achieves 2.55 and 2.31 in terms of fluency and coherence, **GPT-Anno** achieves 2.54 and 2.35 and **SDDS** achieves 2.73 and 2.57. The kappa statistics are 0.53 and 0.46 for fluency and coherence, and that indicates moderate agreement between annotators. We also conduct the paired student t-test between SDDS and GPT-Anno and obtain $p < 0.05$ for both metrics. From this experiment, we find that SDDS outperforms the baselines in both metrics, which demonstrates the SDDS can generate fluent summaries with correct facts. A concrete example is shown in Table 6.

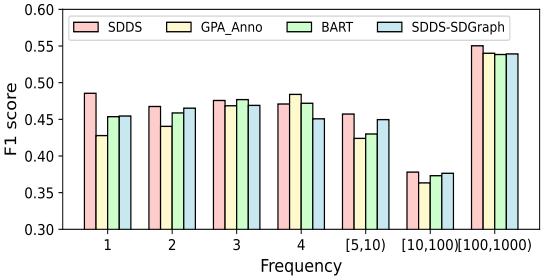

Figure 3: F-measure of words by frequency bucket measured on SAMSum

Efficiency Evaluation Since the construction of static graphs is non-parametric and can be precomputed, the additional training and inference latency is negligible. The training time for BART is 2.15 hours and SDDS is 2.89 hours. The inference speed for BART is 6.87 samples/second and that for SDDS is 6.55. All experiments are conducted on the same computing platform.

Method	R-1	R-2	R-L
SDDS-SDGraph	53.46	28.65	54.78
SDDS-Static	53.68	28.69	55.09
SDDS-Dyna	53.53	28.44	54.85
SDDS w/o Graph	52.91	28.36	54.36
SDDS Simp. Sta.	53.87	29.20	55.51
SDDS	**54.97**	**30.01**	**56.27**

Table 3: Results of ablation study on SAMSum.

6.2 Ablation Study

To prove the effectiveness of each module, we conduct ablation studies that gradually remove each key module in SDDS, and form 5 baseline methods: (1) **SDDS-SDGraph** use relational graph convolutional networks (Schlichtkrull and et al, 2018) to capture high-level hidden features considering different types of edge, and replace the Static-Dynamic Graph (SDG) module proposed in our method; (2) **SDDS-Dyna** only uses the static graph and removes the static-dynamic graph fusion module; (3) **SDDS-Static** only uses the dynamic graph and removes the static-dynamic graph fusion module; (4) **SDDS w/o Graph** does not use any graph model or dialogue structure information, and the decoder directly attends to the utterance representation $\mathbf{U}$ (calculated in Equation 1) instead of attending to graph node representations $\{\mathbf{g_1}, \cdots, \mathbf{g_{L^d}}\}$ as in SDDS; (5) **SDDS Simp. Sta.** verifies the effectiveness of using our proposed 1×1 convolu-

tional layer (shown in Equation 7) to fuse the static graphs, which simply concatenates the adjacent matrixes as $\mathcal{G}^s$.

The results are shown in Table 3. All ablation models perform worse than SDDS in terms of R-1/2/L, which demonstrates the preeminence of SDDS. From the table, we can find that the graph module contributes the most, which demonstrates the necessity of incorporating structural information into the dialogue summarization task. Although the SDDS-SDGraph uses the expressive RGCN to incorporate the dialogue structure information, it is still 2.34% and 1.94% worse than the SDDS in terms of R-1 and R-L scores. Since SDDS Simp. Sta. cannot conduct cross-graph information fusion, it is 1.56% worse than the SDDS in terms of R-1.

Method	R-1	R-2	R-L
w/o Discourse	54.34	29.59	55.84
w/o Keywords	54.47	29.49	55.76
w/o Speak. Rela.	54.12	29.09	55.47
w/o Utter. Posi.	54.08	29.27	55.59
SDDS	**54.97**	**30.01**	**56.27**

Table 4: Importance of different static graphs.

Method	R-1	R-2	R-L
w/ Posi. Emb	54.33	29.13	55.70
w/ Sin. Emb	54.28	29.47	55.69
SDDS	**54.97**	**30.01**	**56.27**

Table 5: Different positional encoding methods.

6.3 On the Different Static Graphs

To evaluate the contribution of each type of static graph, we ablate each static graph, and the results are shown in Table 4. We can find that the utterance position information contributes most to the final performance which demonstrates utterance position can help the model to understand the structure when summarizing the dialogue. Although the discourse parsing graph is an intuitive way to model the dialogue structures and has been widely used in previous dialogue summarization methods (Chen and Yang, 2021; Feng et al., 2021a), it only contributes 0.68% R-1 score compared to the SDDS which is lower than the speaker relation and utterance position. Compare with the model `SDDS-Static` which only uses the dynamic graph module, we can find that the models in Table 4 are all better than `SDDS-Static`. This phenomenon demonstrates the effectiveness of using pre-computed graph structures since it brings human prior knowledge into the dialogue model and future advances in dialogues structure modeling would further benefit SDDS.

#1 Matt: Hey! I got my ticket for Dawid Podsiadlo!!! So stoked!	
#2 Thomas: Whooaa that's great!!	
#3 Matt: I will see you there then!	
#4 Thomas: Yes for sure	
#5 Matt: For sure. Who you're going with?	
#6 Thomas: by myself for now.	
#7 Matt: I might ask a few more people if they're coming :)	
#8 Thomas: Maria was interested I think. But I am not sure. i will ask	
BART	Matt got his ticket for Dawid Podsiadlo's concert. Thomas is going with Maria.
GPT-Anno	Matt got a ticket for Dawid Podsiadlo. He will see Thomas and Maria there.
SDDS	Matt and Thomas are going to Dawid Podsiadlo.
Reference	**Matt got a ticket for Dawid Podsiadlo's concert. Thomas is going, too.**

Table 6: Example of the generated summary by SDDS and other models. Text in red denotes the wrong fact. In the discourse parsing graph, there is no edge between #7 and #8, while our dynamic graph assigns a high weight for the edge between #7 and #8 which captures "whether Maria will go to concert".

6.4 On the Positional Encoding

In the previous section, we can find that the utterance positional static graph contributes most to the final performance in Table 4. In this section, we also compare our positional encoding methods with two commonly used variants: (1) **w/ Posi. Emb**: uses a trainable matrix as the positional embedding of each utterance (Gehring et al., 2017; Lewis et al., 2020) (2) **w/ Sin. Emb**: uses the static sinusoidal function to form a positional encoding vector (Vaswani et al., 2017). From Table 5, we can find that these two methods perform worse than our proposed SDDS. This phenomenon verifies the effectiveness of fusing the positional information into utterance relationships in the static graph.

7 Conclusion

In this paper, we first investigate the limitation of the current static graph-based dialogue summarization methods and propose a Static-Dynamic graph-based Dialogue Summarization method (SDDS). It contains two modules, a static graph module and a dynamic graph module. The former injects human prior into the summarization model and the latter encodes the implicit knowledge from a pre-trained language model. By fusing these two kinds of graphs with a fine-grained 1×1 convolution, SDDS could adaptively adjust the graph weight and

learn the graph structure in an end-to-end learning fashion from the supervision of the summarization task. To validate the effectiveness of SDDS, we conduct extensive experiments on three public dialogue summarization datasets (SAMSum, MediaSum, and DialogSum) and observe significant improvement over strong baselines. We also carefully examine each key component and gives a detailed analysis of SDDS for future research.

Limitations

We discuss the limitations of SDDS as follows:

(1) Although we propose a general framework for dialogue summarization by incorporating both static and dynamic graphs, we only adopt four static graphs to model the dialogue structure. Since dialogue structure modeling is still an active research direction, we believe future advances would further benefit our framework.

(2) Despite the strong performance achieved by SDDS across three dialogue summarization datasets, we use a pre-trained language model as the backbone of our proposed method, as a consequence, we can not go beyond the limitation of the maximum sequence length of the PLM for the dialog summarization scenario like meeting summarization so it remains a future challenge for dialog summarization in the extremely long format.

Ethical Consideration

The dialogue data would inevitably contain private information about the interlocutors. We take careful consideration of this problem: (1) all data in our experiments are publicly available and anonymized by the original dataset provider. The license for SAMSum dataset is *CC BY-NC-ND 4.0* and for DialogSum *MIT License*. For MediaSum, it adheres to only-for-research-purpose guideline from the National Public Radio; (2) we do not use online user data to train our model and we would use an additional rule-based system to double-check whether our model output contains harmful and prejudicial discrimination when we use it for production.

Acknowlegement

This work was supported by the National Natural Science Foundation of China (NSFC Grant No. T2293773 & No. 62122089 & No. 61876196), the National Key Research and Development Program of China (No. 2020AAA0106600), Beijing Outstanding Young Scientist Program (NO. BJJWZYJH012019100020098), and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. Rui Yan is also supported by Beijing Academy of Artificial Intelligence (BAAI).

References

Hangbo Bao and et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *ICML*.

J. Carletta and et al. 2005. The ami meeting corpus: A pre-announcement. In *MLMI*.

Bingkun Chen, Shaobing Dai, Shenghua Zheng, Lei Liao, and Yang Li. 2021a. Dsbert: Unsupervised dialogue structure learning with bert. *arXiv preprint arXiv:2111.04933*.

Jiaao Chen, Mohan Dodda, and Diyi Yang. 2022. Human-in-the-loop abstractive dialogue summarization. *arXiv preprint arXiv:2212.09750*.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *EMNLP*.

Jiaao Chen and Diyi Yang. 2021. Structure-aware abstractive conversation summarization via discourse and action graphs. In *NAACL*.

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

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021b. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Xin Cheng, Shen Gao, Lemao Liu, Dongyan Zhao, and Rui Yan. 2022. Neural machine translation with contrastive translation memories. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3591–3601, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xin Cheng, Shen Gao, Yuchi Zhang, Yongliang Wang, Xiuying Chen, Mingzhe Li, Dongyan Zhao, and Rui Yan. 2023a. Towards personalized review summarization by modeling historical reviews from customer and product separately. *arXiv preprint arXiv:2301.11682*.

Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2023b. Lift yourself up: Retrieval-augmented text generation with self memory. *arXiv preprint arXiv:2305.02437*.

Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019a. Metapath-guided heterogeneous graph neural network for intent recommendation. In *KDD*.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019b. Graph neural networks for social recommendation. In *WWW*.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2022. A survey on dialogue summarization: Recent advances and new frontiers. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5453–5460. ijcai.org.

Xiachong Feng, Xiaocheng Feng, Bing Qin, Xinwei Geng, and Ting Liu. 2021a. Dialogue discourse-aware graph convolutional networks for abstractive meeting summarization. In *IJCAI*.

Xiachong Feng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2020. Incorporating commonsense knowledge into abstractive dialogue summarization via heterogeneous graph networks. In *Arxiv*.

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021b. Language model as an annotator: Exploring dialogpt for dialogue summarization. In *ACL*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-up abstractive summarization.

Zhichao Geng, Ming Zhong, Zhangyue Yin, Xipeng Qiu, and Xuan-Jing Huang. 2022a. Improving abstractive dialogue summarization with speaker-aware supervised contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6540–6546.

Zhichao Geng, Ming Zhong, Zhangyue Yin, Xipeng Qiu, and Xuanjing Huang. 2022b. Improving abstractive dialogue summarization with speaker-aware supervised contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6540–6546, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *EMNLP*.

Chih-Wen Goo and Yun-Nung Chen. 2018. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 735–742. IEEE.

Iryna Gurevych and Michael Strube. 2004. Semantic similarity applied to spoken dialogue summarization. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 764–770.

Adam L. Janin and et al. 2003. The icsi meeting corpus. *ICASSP '03*.

Pin Jiang and Yahong Han. 2020. Reasoning with heterogeneous graph alignment for video question answering. In *AAAI*.

Zhuoren Jiang, Yue Yin, Liangcai Gao, Yao Lu, and Xiaozhong Liu. 2018. Cross-language citation recommendation via hierarchical representation learning on heterogeneous graph. In *SIGIR*.

Daniel Jurafsky and James H Martin. 2000. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Changqun Li, Linlin Wang, Xin Lin, Gerard de Melo, and He Liang. 2022. Curriculum prompt learning with self-training for abstractive dialogue summarization. In *Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics.

Jinpeng Li, Yingce Xia, Xin Cheng, Dongyan Zhao, and Rui Yan. 2023. Learning disentangled representation via domain adaptation for dialogue summarization. In *Proceedings of the ACM Web Conference 2023*, pages 1693–1702.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019. Automatic dialogue summary generation for customer service. In *KDD*.

Junpeng Liu, Yanyan Zou, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Caixia Yuan, and Xiaojie Wang. 2021a. Topic-aware contrastive learning for abstractive dialogue summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1229–1243, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Siwei Liu, Iadh Ounis, Craig Macdonald, and Zaiqiao Meng. 2020. A heterogeneous graph neural model for cold-start recommendation. In *SIGIR*.

Zhengyuan Liu, Ke Shi, and Nancy Chen. 2021b. Coreference-aware dialogue summarization. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 509–519, Singapore and Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Boer Lyu, Lu Chen, Su Zhu, and Kai Yu. 2021. Let: Linguistic knowledge enhanced graph transformer for chinese short text matching. In *AAAI*.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2639–2649. Association for Computational Linguistics.

Mathieu Ravaut, Shafiq Joty, and Nancy F Chen. 2022. Towards summary candidates fusion. *arXiv preprint arXiv:2210.08779*.

Michael Schlichtkrull and et al. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multi-party dialogues. In *AAAI*.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *CICLing*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *ACL*.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. Investigating BERT's knowledge of language: Five analysis methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Yang Wei. 2012. Document summarization method based on heterogeneous graph. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 1285–1289. IEEE.

Chien-Sheng Wu, Linqing Liu, Wenhao Liu, Pontus Stenetorp, and Caiming Xiong. 2021. Controllable abstractive dialogue summarization with sketch supervision. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5108–5122, Online. Association for Computational Linguistics.

Yongji Wu, Defu Lian, Shuowei Jin, and Enhong Chen. 2019a. Graph convolutional networks on user mobility heterogeneous graphs for social relationship inference. In *IJCAI*.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019b. Relation-aware entity alignment for heterogeneous knowledge graphs. In *IJCAI*.

Xiaoyu Yang, Yuefei Lyu, Tian Tian, Yifei Liu, Yudong Liu, and Xi Zhang. 2020. Rumor detection on social media with graph structured adversarial learning. In *IJCAI*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Shiyue Zhang, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal. 2021. Emailsum: Abstractive email thread summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6895–6909. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yizhe Zhang and et al. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Lulu Zhao, Weiran Xu, and Jun Guo. 2020. Improving abstractive dialogue summarization with graph structures and topic words. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 437–449.

Lulu Zhao, Weihao Zeng, Weiran Xu, and Jun Guo. 2021a. Give the truth: Incorporate semantic slot into abstractive dialogue summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2435–2446, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lulu Zhao, Fujia Zheng, Keqing He, Weihao Zeng, Yuejie Lei, Huixing Jiang, Wei Wu, Weiran Xu, Jun Guo, and Fanyu Meng. 2021b. Todsum: Task-oriented dialogue summarization with state tracking. *arXiv preprint arXiv:2110.12680*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. Mediasum: A large-scale media interview dataset for dialogue summarization. In *NAACL*.

A Appendix

A.1 Speaker Relation Graph

We use a sliding window around each utterance, and count the frequency of occurrence for each speaker in this sliding window. Figure 4 gives an example to illustrate this method, and we obtain a speaker interaction frequency matrix $\hat{\mathcal{G}}_s^s \in \mathbb{N}^{|S|,|S|}$. Algorithm 1 illustrates this method, and we obtain a speaker interaction frequency matrix $\hat{\mathcal{G}}_s^s \in \mathbb{N}^{|S|,|S|}$.

A.2 Implementation Details

We implement our experiments using Pytorch (Paszke et al., 2019) on an NVIDIA RTX 3090 GPU. The batch size is set to 16, and we use the gradient accumulation to simulate a large batch size. We pad or cut input utterances to contain exactly 200 words, and the maximum decoding length is set to 100. We initialize BART in our model with BART_{Large}[†] which has 16 attention heads, 1024 hidden size and 12 Transformer layers for encoder and decoder respectively. In our graph transformer, we use 4 self-attention layers with 1024 hidden size and 8 attention head. We use AdamW optimizer (Loshchilov and Hutter, 2019) as our optimizing algorithm and employ beam search with size 5 to generate more fluency summary.

A.3 Dataset Statistics

We list some key statistics of these datasets in Table 7. From this table, we can find that the MediaSum-NPR dataset has more speakers, training samples, and longer dialogue context than the other datasets. Note that, in DialogSum, there are three reference summaries for each data sample, and we use multiple references in the evaluation.

	SAMSum	MediaSum-NPR	DialogSum
# of training samples	14,732	47,370	12,460
# of test samples	819	1,060	500
# of validation samples	818	990	500
Avg. turns of dialogue	9.9	24.2	9.49
Avg. speakers of dialogue	2.2	4.0	2.01
Avg. words of summary	20.3	14.4	22.87

Table 7: Dataset Statistics for three benchmark datasets: SAMSum, MediaSum-NPR and DialogSum.

A.4 Evaluation Metrics

For evaluation metrics, following existing dialogue summarization papers (Feng et al., 2021b), we adopt ROUGE score (Lin, 2004), which is widely applied for summarization evaluation (Chen et al., 2018). The ROUGE metrics compare generated summary with the reference summary by computing overlapping lexical units, including ROUGE-1 (unigram), ROUGE-2 (bi-gram), and ROUGE-L (longest common subsequence). Following existing dialogue summarization papers (Feng et al., 2021b), we use py-rouge[‡] as the implementation of ROUGE score. Since only using automatic evaluation metrics can be misleading (Stent et al., 2005), we also use the embedding based evaluation method and conduct the human evaluation. We employ the BERTScore (Zhang et al., 2020b), BARTScore (Yuan et al., 2021) and MoverScore (Zhao et al., 2019) as the embedding based evaluation. For human evaluation, three well-educated annotators are invited to judge 200 randomly sampled summaries. The statistical significance of two runs is tested using a two-tailed paired t-test and is denoted using $^{\blacktriangle}$(or $^{\blacktriangledown}$) for strong significance for $\alpha = 0.01$.

Algorithm 1 Algorithm of speaker relation construction.

Input: Dialog Context with L_d utterances
Output: Speaker relation $\mathcal{G}_s^s \in \mathbb{R}^{L_d,L_d}$

Let $\hat{\mathcal{G}}_s^s \in \mathbb{N}^{|S|,|S|} = \mathbf{0}$.
$\alpha(u_j)$ = speaker index of u_j
for each u_i in D
  for each u_j in sliding window of u_i
    $\hat{\mathcal{G}}_s^s(\alpha(u_i), \alpha(u_j)) = \hat{\mathcal{G}}_s^s(\alpha(u_i), \alpha(u_j)) + 1$
$\tilde{\mathcal{G}}_s^s = \text{softmax}_r(\hat{\mathcal{G}}_s^s) \times \text{softmax}_c(\hat{\mathcal{G}}_s^s)$
for each i in $\{1, \ldots, L_d\}$
  for each j in $\{1, \ldots, L_d\}$
    $\mathcal{G}_s^s(i, j) = \hat{\mathcal{G}}_s^s(\alpha(u_i), \alpha(u_j))$
return $\mathcal{G}_s^s(i, j)$

[†] https://huggingface.co/facebook/bart-large

[‡] https://pypi.org/project/py-rouge

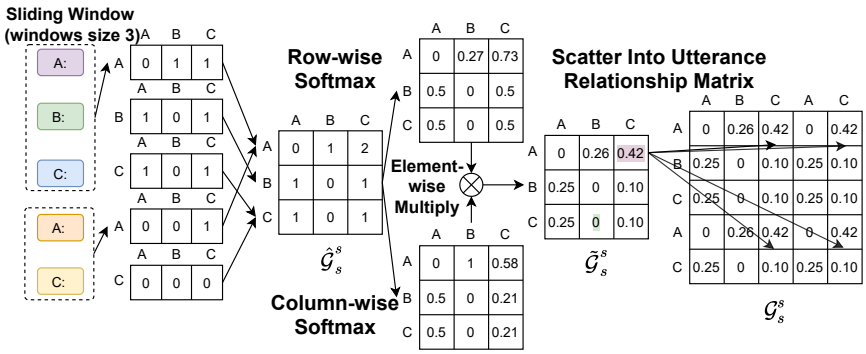

Figure 4: An example of speaker relation graph construction.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
the last section

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
the first two sections

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

in section 6

☑ B1. Did you cite the creators of artifacts you used?
in section 5

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
in the ethical consideration section

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
in section 5 and appendix

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
in the appendix

C ☑ Did you run computational experiments?

in section 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
in appendix

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
in section 5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
in section 6

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
in appendix

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

section 6

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
section 6

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
section 6

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.