Dialogue Discourse-Aware Graph Model and Data Augmentation for Meeting Summarization

Xiachong Feng, **Xiaocheng Feng**, **Bing Qin**, **Xinwei Geng**
Harbin Institute of Technology, China
{xiachongfeng, xcfeng, bqin, xwgeng}@ir.hit.edu.cn

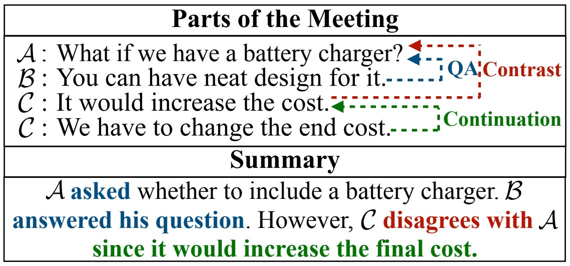
Abstract

Meeting summarization is a challenging task due to its dynamic interaction nature among multiple speakers and lack of sufficient training data. Existing methods view the meeting as a linear sequence of utterances while ignoring the diverse relations between each utterance. Besides, the limited labeled data further hinders the ability of data-hungry neural models. In this paper, we try to mitigate the above challenges by introducing dialogue-discourse relations. First, we present a **D**ialogue **D**iscourse-**A**ware **M**eeting **S**ummarizer (DDAMS) to explicitly model the interaction between utterances in a meeting by modeling different discourse relations. The core module is a relational graph encoder, where the utterances and discourse relations are modeled in a graph interaction manner. Moreover, we devise a **D**ialogue **D**iscourse-**A**ware **D**ata **A**ugmentation (DDADA) strategy to construct a pseudo-summarization corpus from existing input meetings, which is 20 times larger than the original dataset and can be used to pretrain DDAMS. Experimental results on AMI and ICSI meeting datasets show that our full system can achieve SOTA performance Our codes and outputs are available at: https://github.com/xcfcode/DDAMS/.

1 Introduction

Meeting summarization aims to distill the most important information from a recorded meeting into a short textual passage, which could be of great value to people by providing quick access to the essential content of past meetings [Gurevych and Strube, 2004]. An example is shown in Figure 1, which includes 3 speakers ($\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$) and their corresponding utterances, as well as a human-written summary.

Recently, neural-based sequence-to-sequence methods have become the mainstream mode for generating meeting summaries via accurately modeling the refined semantic representation of the original meeting text. Goo and Chen [2018] incorporate utterance-level dialogue acts to enhance the representation for each utterance. Li *et al.* [2019] consider topics as a structure information to enrich the meeting representation. However, we claim that the current research suffers from



Parts of the Meeting

$\mathcal{A}$: What if we have a battery charger?
$\mathcal{B}$: You can have neat design for it.
$\mathcal{C}$: It would increase the cost.
$\mathcal{C}$: We have to change the end cost.

Summary

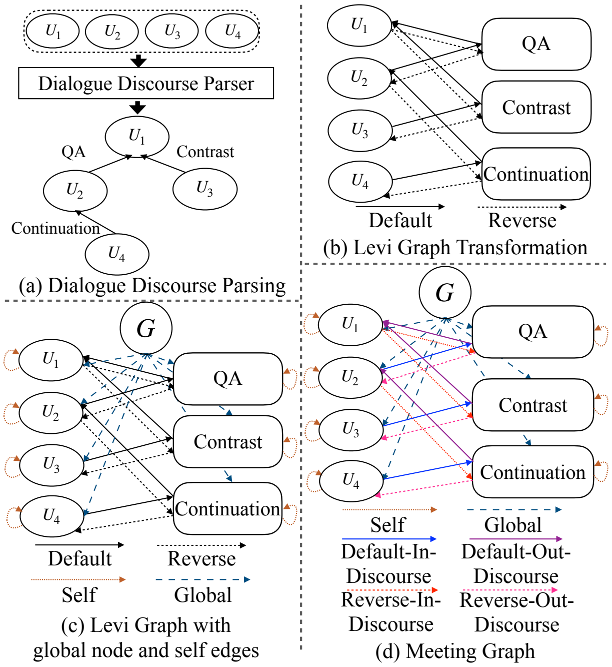
$\mathcal{A}$ **asked** whether to include a battery charger. $\mathcal{B}$ **answered his question**. However, $\mathcal{C}$ **disagrees with** $\mathcal{A}$ **since it would increase the final cost.**

Figure 1: An example of a meeting with its corresponding summary. *QA*, *Contrast* and *Continuation* are dialogue discourse relations, which explicitly show the interaction between utterances.

two problems. One is sequential text modeling. A meeting is a dynamic information exchange flow, which is informal, verbose, and less structured than traditional documents [Sacks *et al.*, 1978]. But all previous works adopt a sequential modeling strategy for encoding the meeting, hindering the exploration of inherently rich interactive relations between utterances, which makes the meeting modeling inadequate. The other one is the lack of sufficient training data. As we know, the scale of corpus is significant to the performance of neural models, the data size of meeting summarization is only one-thousandth of the traditional news summarization (e.g. 137 meetings in AMI v.s. 312K articles in CNN/DM).

To address the above concerns, we draw support from dialogue discourse, a dialogue-specific structure, which can provide pre-defined relations between utterances [Stone *et al.*, 2013]. As shown in Figure 1, *QA*, *Contrast* and *Continuation* are three dialogue discourse relations, which explicitly show the information flow and interactions between utterances. After knowing these relations, we can generate better summaries. Furthermore, we find that a question often sparks a discussion in a meeting. As shown in Figure 1, the discussion about "design" and "cost" surrounds the salient term "battery charger" in the question. Therefore, we assume that a question tends to contain core topics or key concepts that can be used as a pseudo summary for subsequent discussion.

In this paper, we propose a **D**ialogue **D**iscourse-**A**ware **M**eeting **S**ummarizer (DDAMS, see §3) for modeling the dialogue discourse relations among utterances. Specifically, we first convert meeting utterances with discourse relations into a meeting graph where utterance vertices interact with rela-



(a) Dialogue Discourse Parsing

(b) Levi Graph Transformation

(c) Levi Graph with global node and self edges

(d) Meeting Graph

Figure 2: Illustration of meeting graph construction process.

tion vertices. Then we design a graph-to-sequence framework to generate meeting summaries. Besides, to tackle the insufficient training problem, we devise a **D**ialogue **D**iscourse-**A**ware **D**ata **A**ugmentation (DDADA, see §4) strategy, which constructs a pseudo-summarization corpus from existing input meetings. In detail, we use the *QA* discourse relation to identify the question as pseudo-summary, then we view subsequent utterances with associated discourse relations as a pseudo-meeting. Finally, we can construct a pseudo-summarization corpus, which is 20 times larger than the original dataset and can be used to pretrain DDAMS.

We conduct various experiments on the widely used AMI [Carletta *et al.*, 2005] and ICSI [Janin *et al.*, 2003] datasets. The results show the effectiveness of DDAMS method and DDADA strategy. In summary, (1) We make the first attempt to successfully explore dialogue discourse to model the utterances interactions for meeting summarization; (2) We devise a dialogue discourse-aware data augmentation strategy to alleviate the data insufficiency problem; (3) Extensive experiments show that our model achieves SOTA performance.

2 Preliminaries

In this section, we describe the task definition and give a brief introduction to dialogue discourse.

2.1 Task Definition

Meeting summarization aims at producing a summary $\mathcal{Y}$ for the input meeting $\mathcal{U}$, where $\mathcal{U}$ consists of $|\mathcal{U}|$ utterances $[u_1, u_2, ...u_{|\mathcal{U}|}]$ and $\mathcal{Y}$ consists of $|\mathcal{Y}|$ words $[y_1, y_2, ...y_{|\mathcal{Y}|}]$. The i-th utterance of the meeting can be represented as a sequence of words $u_i = [u_{i,1}, u_{i,2}, ...u_{i,|u_i|}]$, where $u_{i,j}$ denotes the j-th word of i-th utterance. Each utterance u_i associates with a speaker $p_i \in \mathbb{P}$, $\mathbb{P}$ being a set of participants.

2.2 Dialogue Discourse

Dialogue discourse indicates relations between discourse units in a conversation (i.e. utterances in a meeting). This dependency-based structure allows relations between non-adjacent utterances that is applicable for multi-party conversations [Li *et al.*, 2021]. There are 16 discourse relations [Asher *et al.*, 2016] in total: *comment, clarification-question, elaboration, acknowledgment, continuation, explanation, conditional, QA, alternation, question-elaboration, result, background, narration, correction, parallel, contrast*.

3 Dialogue Discourse-Aware Meeting Summarizer

In this section, we present our Dialogue Discourse-Aware Meeting Summarizer (DDAMS) that consists of four components: a meeting graph, a vertex representation module, a graph encoder, and a pointer decoder, as shown in Figure 3.

3.1 Meeting Graph

Given meeting utterances, we first use a SOTA dialogue discourse parser [Shi and Huang, 2019] to get discourse relations, one relation connects from one utterance to another one with a relation type, as shown in Figure 2(a).

Then, we perform the Levi graph transformation, which turns labeled edges into additional vertices [Gross *et al.*, 2013]. Through this transformation, we can model discourse relations explicitly and update the utterance and discourse vertices simultaneously. There are two types of edges in the Levi graph: *default* and *reverse*. For example, an edge (U_3, *Contrast*, U_1) in the original graph becomes (U_3, *default*, *Contrast*) and (*Contrast*, *default*, U_1) in the Levi graph, as shown in Figure 2(b).

Furthermore, to aggregate non-local information, a global vertex is added that connects all vertices by *global* edges and will be used to initialize the decoder. We also add *self* edges to aggregate self-information, as shown in Figure 2(c).

Note that different types of vertices may have different features that fall in different space [Beck *et al.*, 2018]. Specifically, previous works ignore the type of source and target vertices and use the same type of edge to pass information, which may reduce the effectiveness of discourse information. Thus, we propose our meeting graph which transforms the *default* edge into *default-in-discourse* and *default-out-discourse* edges and the *reverse* edges into *reverse-in-discourse* and *reverse-out-discourse* edges, as shown in Figure 2(d).

Let $\mathcal{G}_M = (\mathbb{V}_M, \mathbb{E}_M, \mathbb{R}_M)$ denote a meeting graph, with vertices $v_i \in \mathbb{V}_M$ and labeled edges $(v_i, r, v_j) \in \mathbb{E}_M$, where $r \in \mathbb{R}_M$ is the relation type of the edge. For our meeting graph, there are six types of relations, where $\mathbb{R}_M$ becomes *default-in-discourse, default-out-discourse, reverse-in-discourse, reverse-out-discourse, global, self*.

3.2 Vertex Representation

Vertex representation module aims to obtain initial representations for three types of vertices: global vertex, relation vertex and utterance vertex. For global vertex and relation vertices, we get the initial representation $\boldsymbol{h}_i^0$ by

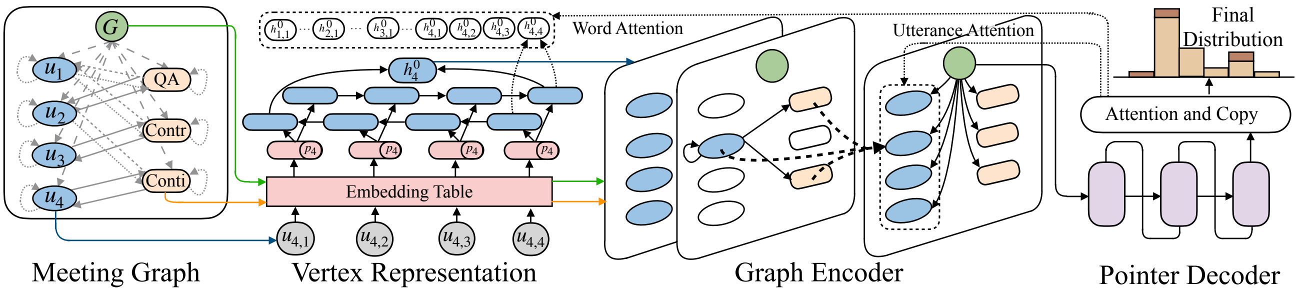


Figure 3: Illustration of our DDAMS model. (1) First, we construct our meeting graph consisting of three types of vertices: global vertex, utterance vertex and discourse relation vertex. (2) Then, the vertex representation module gives each type of vertex an initial representation. (3) Further, the graph encoder performs convolutional computation over the meeting graph based on the relational graph convolutional network. (4) Finally, the pointer decoder attends to the updated utterance representations and the word representations to generate the summary words either from the fixed-length vocabulary or copy from the input.

looking up from an embedding table. For utterance vertices, we employ a BiLSTM[1] as the utterance encoder that encodes the utterance forwardly and backwardly [Qin *et al.*, 2020]. As $\overrightarrow{\boldsymbol{h}_{i,j}} = \text{LSTM}_f\left(\overrightarrow{\boldsymbol{h}_{i,j-1}}, \boldsymbol{e}_{i,j}\right)$, $\overleftarrow{\boldsymbol{h}_{i,j}} = \text{LSTM}_b\left(\overleftarrow{\boldsymbol{h}_{i,j+1}}, \boldsymbol{e}_{i,j}\right)$, $\boldsymbol{h}_{i,j} = [\overrightarrow{\boldsymbol{h}_{i,j}}; \overleftarrow{\boldsymbol{h}_{i,j}}]$, where $\boldsymbol{h}_{i,j}$ and $\boldsymbol{e}_{i,j}$ denote the hidden state and embedding of word $u_{i,j}$. ; denotes concatenation. To involve speaker information, we encode speaker p_i using a one-hot vector and get $\boldsymbol{e}_{i,j}$ by concatenating word embedding and one-hot speaker embedding [Wang *et al.*, 2020]. The representation $\boldsymbol{h}_i^0 = [\overrightarrow{\boldsymbol{h}_{i,|u_i|}}; \overleftarrow{\boldsymbol{h}_{i,1}}]$ is used as input to the graph encoder.

3.3 Graph Encoder

After getting the initial feature $\boldsymbol{h}_i^0$ of each vertex $v_i \in \mathbb{V}_M$, we feed them into the graph encoder to digest the structural information. We use relational graph convolutional networks [Schlichtkrull *et al.*, 2018] to capture high-level hidden features considering different types of edges. The convolutional computation for v_i at the $(l+1)$-th layer takes the representation $\boldsymbol{h}^{(l)}$ at the l-th layer as input can be defined as:

$$\boldsymbol{h}_i^{(l+1)} = \text{ReLU}\left(\sum_{r\in\mathbb{R}_M}\sum_{v_j\in\mathbb{N}_i^r}\frac{1}{|\mathbb{N}_i^r|}\boldsymbol{W}_r^{(l)}\boldsymbol{h}_j^{(l)}\right) \quad (1)$$

where $\mathbb{N}_i^r$ denotes the set of neighbors of vertex v_i under relation r and $\boldsymbol{W}_r^{(l)}$ denotes relation-specific learnable parameters at the l-th layer.

However, uniformly accepting information from different discourse relations is not suitable for identifying important discourse. Thus, we use the gate mechanism [Marcheggiani and Titov, 2017] to control the information passing:

$$g_j^{(l)} = \text{sigmoid}\left(\boldsymbol{W}_{g,r}^{(l)}\boldsymbol{h}_j^{(l)}\right) \quad (2)$$

where $\boldsymbol{W}_{g,r}^{(l)}$ denotes a learnable parameter under relation type r at the l-th layer. Equipped with the gate mechanism, the convolutional computation can be defined as:

$$\boldsymbol{h}_i^{(l+1)} = \text{ReLU}\left(\sum_{r\in\mathbb{R}_M}\sum_{v_j\in\mathbb{N}_i^r} g_j^{(l)}\frac{1}{|\mathbb{N}_i^r|}\boldsymbol{W}_r^{(l)}\boldsymbol{h}_j^{(l)}\right) \quad (3)$$

3.4 Pointer Decoder

We use a standard LSTM decoder with attention and copy mechanism to generate the summary [Bahdanau *et al.*, 2015; See *et al.*, 2017]. The global representation is used to initialize the decoder. At each step t, the decoder receives the word embedding of previous word and has decoder state $\boldsymbol{s}_t$. The attention distribution is calculated as in [Luong *et al.*, 2015]. We consider both word-level and utterance-level attention. The word-level context vector $\boldsymbol{h}_t^{wl}$ is computed as:

$$\begin{aligned} e_{i,j}^t &= \boldsymbol{s}_t^\top \boldsymbol{W}_a \boldsymbol{h}_{i,j}^0 \\ \boldsymbol{a}^t &= \text{softmax}(\boldsymbol{e}^t) \\ \boldsymbol{h}_t^{wl} &= \sum_i\sum_j a_{i,j}^t \boldsymbol{h}_{i,j}^0 \end{aligned} \quad (4)$$

where $\boldsymbol{W}_a$ is a learnable parameter and $\boldsymbol{h}_{i,j}^0$ is obtained from utterance encoder for $u_{i,j}$. The utterance-level context vector $\boldsymbol{h}_t^{ul}$ is calculated similarly to the word-level context vector, except that we use the final outputs of the graph encoder $\boldsymbol{h}_i^{(l)}$ which represent utterances to calculate the attention distribution. The final context vector is the concatenation of word-level and utterance-level context vector $\boldsymbol{h}_t^* = [\boldsymbol{h}_t^{wl}; \boldsymbol{h}_t^{ul}]$, which is then used to calculate generation probability and the final probability distribution [See *et al.*, 2017].

3.5 Training Objective

We use maximum likelihood estimation to train our model. Given the ground truth summary $\mathcal{Y}^* = [y_1^*, y_2^*, ..., y_{|\mathcal{Y}^*|}^*]$ for an input meeting $\mathcal{U}$. We minimize the negative log-likelihood of the target words sequence:

$$\mathcal{L} = -\sum_{t=1}^{|\mathcal{Y}^*|}\log p\left(y_t^* | y_1^* \dots y_{t-1}^*, \mathcal{U}\right) \quad (5)$$

[1] We have also tried Transformer [Vaswani *et al.*, 2017] as our backbone model. However, it has too many parameters which can easily over-fit on the small meeting dataset.

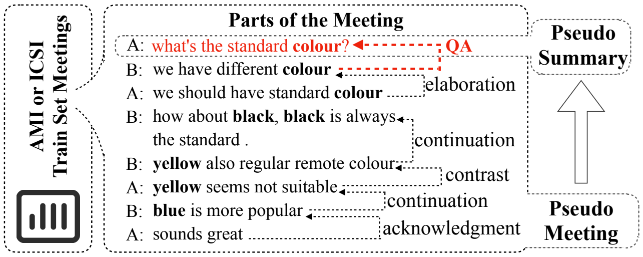

AMI or ICSI Train Set Meetings
Parts of the Meeting
A: what's the standard colour?
QA
B: we have different colour
A: we should have standard colour
elaboration
B: how about black, black is always the standard .
continuation
B: yellow also regular remote colour
contrast
A: yellow seems not suitable
B: blue is more popular
continuation
A: sounds great
acknowledgment
Pseudo Summary
Pseudo Meeting

Figure 4: Illustration of how to construct a pseudo meeting-summary pair. Given a meeting from the original meeting train set, we use *QA* discourse relation to identify the question in the meeting. Then, the subsequent discussion with discourse relations becomes a pseudo meeting and the question becomes a pseudo summary.

4 Dialogue Discourse-Aware Data Augmentation

In this section, we introduce Dialogue Discourse-Aware Data Augmentation strategy (DDADA), which constructs a pseudo-summarization corpus from the original input meetings based on the *QA* discourse relation.

4.1 Pseudo-summarization Corpus Construction

Given a meeting and its corresponding discourse relations, we find a question often sparks a discussion and contains salient terms or concepts expressed in the discussion. As shown in Figure 4, speaker A asked "What's the standard colour?", other participants start to discuss this small topic. Thus, we can view the discussion as a pseudo meeting and the question as a pseudo summary of this pseudo meeting[2].

According to the above observation, we collect pseudo-summarization data from the original meeting summarization dataset where the question identified by *QA* discourse serves as the pseudo summary and N utterances after the question with their associated discourse relations serve as the pseudo meeting[3]. Note that there are some uninformative and normal questions such as "what is this here", which are not suitable for pseudo-summarization corpus construction. Thus, we filter out questions that contain no noun and adjective to make the pseudo data more realistic. Table 1 shows the statistics for the pseudo-summarization corpus.

4.2 Pretraining

Given the pseudo corpus, we first pretrain DDAMS through a question generation pretraining objective, which pretrains DDAMS to generate questions conditioning on subsequent discussions. Afterward, we fine-tune DDAMS on the meeting summarization dataset. Narayan *et al.* [2020] also reveal that treating question generation as the pretraining objective can help downstream tasks in identifying important contents in the open domain. Our motivations are two-fold: (1) we can potentially augment the training data; (2) since the pseudo data is constructed from the meeting summarization dataset (in-domain), the pre-training can give the model a warm start.

	AMI Pseudo Corpus	**ICSI Pseudo Corpus**
# of Original Data	97	53
# of Pseudo Data	1539	1877
Avg.Tokens	124.44	107.44
Avg.Sum	13.18	11.97

Table 1: Pseudo-summarization corpus statistics. "# of Original Data" means the number of original meetings in the train set, "# of Pseudo Data" means the number of pseudo meeting-summary pairs, "Avg.Tokens" means the average length of pseudo meetings and "Avg.Sum" means the average length of pseudo summaries.

5 Experiments

Datasets We experiment on AMI [Carletta *et al.*, 2005] and ICSI [Janin *et al.*, 2003] datasets. We preprocess the dataset into train, valid and test sets for AMI (97/20/20) and ICSI (53/25/6) separately following Shang *et al.* [2018]. We get discourse relations for one meeting based on the SOTA dialogue discourse parser, namely Deep Sequential [Shi and Huang, 2019], which is trained on the STAC corpus [Asher *et al.*, 2016].

Evaluation Metrics We adopt ROUGE [Lin, 2004] for evaluation and obtain the F_1 scores for ROUGE-1, ROUGE-2, and ROUGE-L between the ground-truth and the generated summary respectively.

Baseline Models TextRank [Mihalcea and Tarau, 2004] is a graph-based extractive method. **SummaRunner** [Nallapati *et al.*, 2017] is an extractive method based on hierarchical RNN network. **UNS** [Shang *et al.*, 2018] is an unsupervised abstractive method that combines several graph-based approaches. **Pointer-Generator** [See *et al.*, 2017] is an abstractive method equips with copy mechanism. **HRED** [Serban *et al.*, 2016] is a hierarchical Seq2Seq model. **Sentence-Gated** [Goo and Chen, 2018] is an abstractive method that incorporates dialogue acts by the sentence-gated mechanism. **TopicSeg** [Li *et al.*, 2019] is an abstractive method using a hierarchical attention mechanism at three levels (topic, utterance, word).[4] **HMNet** [Zhu *et al.*, 2020] is an abstractive method that employs a hierarchical model and incorporates part-of-speech and entity information, which is also pre-trained on large-scale news summarization dataset[5].

5.1 Automatic Evaluation

As shown in Table 2, our model DDAMS outperforms various baselines, which shows the effectiveness of dialogue discourse. By pre-training on pseudo-summarization data, our model DDAMS +DDADA can further boost the performance

[2] In our preliminary experiments, we randomly sample 100 pseudo-summarization instances and observe that 68% of questions can cover important information, such as salient terms and concepts.

[3] We conduct experiments with $N \in [6:12]$. Finally, we choose N=10 which performs the best.

[4] [Li *et al.*, 2019] also proposed TopicSeg+VFOA by incorporating vision features in a multi-modal setting. In this paper, we compare our model with baselines using only textual features.

[5] Zhu *et al.* [2020] directly use news summarization dataset to pre-train their model, which can not be applied to our model, since our DDAMS needs dialogue discourse information.

		AMI			ICSI		
	Model	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
Extractive	TextRank [Mihalcea and Tarau, 2004]	35.19	6.13	15.70	30.72	4.69	12.97
	SummaRunner [Nallapati *et al.*, 2017]	30.98	5.54	13.91	27.60	3.70	12.52
Abstractive	UNS [Shang *et al.*, 2018]	37.86	7.84	13.72	31.73	5.14	14.50
	Pointer-Generator [See *et al.*, 2017]	42.60	14.01	22.62	35.89	6.92	15.67
	HRED [Serban *et al.*, 2016]	49.75	18.36	23.90	39.15	7.86	16.25
	Sentence-Gated [Goo and Chen, 2018]	49.29	19.31	24.82	39.37	9.57	17.17
	TopicSeg [Li *et al.*, 2019]	51.53	12.23	25.47	-	-	-
	HMNet [Zhu *et al.*, 2020]	52.36	18.63	24.00	**45.97**	10.14	18.54
Ours	DDAMS	51.42	20.99	24.89	39.66	10.09	17.53
	DDAMS + DDADA	**53.15**	**22.32**	**25.67**	40.41	**11.02**	**19.18**
	DDAMS + DDADA (w/o fine-tune)	28.35	4.67	14.92	25.94	4.18	13.92

Table 2: Test set results on AMI and ICSI Datasets, where "R-1" is short for "ROUGE-1", "R-2" for "ROUGE-2", "R-L" for "ROUGE-L". The DDAMS represents the model that is trained only on the meeting dataset. The DDAMS +DDADA means the model that is pre-trained using pseudo-summarization data and then fine-tuned on meeting dataset. DDAMS +DDADA (w/o fine-tune) means the model that is pre-trained using pseudo-summarization data and without fine-tuning on the meeting dataset.[6]

by a large margin, which shows the need for DDADA. Specially, DDAMS +DDADA (w/o fine-tune) still achieves a basic effect, which appears DDAMS +DDADA (w/o fine-tune) can simulate as a summarization model in terms of selecting important information.

5.2 Human Evaluation

To further assess the quality of the generated summaries, we conduct a human evaluation study. We choose two metrics: **relevance** (consistent with the original input) and **informativeness** (preserves the meaning expressed in the ground-truth). We hired five graduates to perform the human evaluation. They were asked to rate each summary on a scale of 1 (worst) to 5 (best) for each metric. The results are shown in Table 3.

	Model	**Relevance**	**Informativeness**
AMI	Ground-truth	4.60	4.56
	Sentence-Gated	3.16	3.60
	HMNet	3.60	3.72
	DDAMS	3.80	3.76
	DDAMS +DDADA	**3.84**	**3.88**
ICSI	Ground-truth	4.76	4.48
	Sentence-Gated	3.32	3.48
	HMNet	3.80	3.52
	DDAMS	3.76	3.28
	DDAMS +DDADA	**3.84**	**3.60**

Table 3: Human evaluation results.

We can see that our DDAMS + DDADA achieves higher scores in both relevance and informativeness than other baselines. Ground truth obtains the highest scores compare with generated summaries indicating the challenge of this task.

5.3 Analysis

Effect of the number of dialogue discourse. We conduct experiments by randomly providing parts of discourse relations to our model DDAMS in the test process. The results are shown in Figure 5. We can see that the more discourse relations, the higher the ROUGE score, which indicates discourse can do good to summary generation. When given no discourse information, our model gets similar scores compared with HRED, which models utterances sequentially.

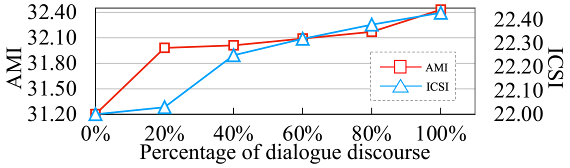


Figure 5: Average ROUGE scores with respect to the number of dialogue discourse relations given at test process.

Effect of the quality of dialogue discourse. We train our DDAMS by using discourse relations that are provided by parsers of different qualities. We can see that the higher the quality of discourse parser, the higher the ROUGE score, which potentially indicates high-quality discourse relations can further improve our model.

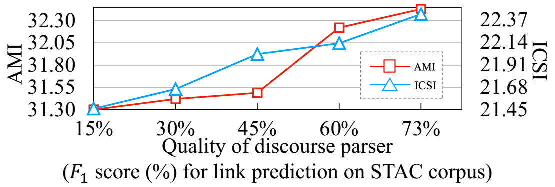


Figure 6: Average ROUGE scores with respect to the quality of discourse parser.

[6] ROUGE-2 of TopicSeg is significantly lower than other baselines, which contains some anomalies. we reproduce the results of UNS and HMNet by the official codes and evaluate them by PyRouge package, the results are consistent with the original paper, which shows the increase of our ROUGE-2 is reasonable.

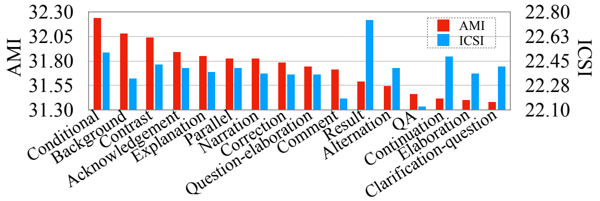


Figure 7: Average ROUGE scores with respect to different types of discourse relations given at test process.

Effect of the type of dialogue discourse. To verify the importance of each discourse relation type, we then test our model by giving different discourse relations. The results are shown in Figure 7. On AMI dataset, the participants take a remote control design project. We can see that *Conditional* and *Background* provide more accurate and adequate information, facilitating summary generation. On ICSI dataset, meetings focus on academic discussion. We can see that the relation *Result* is more important than other relations, we attribute this to the fact that this relation can strongly indicate the outcome of a discussion that covers salient information.

Effect of meeting graph. To verify the effectiveness of designing different types of edges for different types of vertices, we replace our meeting graph with Levi graph (shown in Figure 2(c)), namely DDAMS (w/ Levi graph). The results are shown in Table 4. We can see that taking the type of vertices into consideration, our model DDAMS can get better results.

	Model	R-1	R-2	R-L
AMI	DDAMS	51.42	**20.99**	**24.89**
	DDAMS (w/ Levi graph)	**51.46**	20.75	24.31
ICSI	DDAMS	**39.66**	**10.09**	**17.53**
	DDAMS (w/ Levi graph)	39.20	9.54	17.48

Table 4: Test set results of using meeting graph and Levi graph.

Effect of pseudo-summarization data. To further study the effectiveness of pseudo-summarization data, instead of using questions identified by dialogue discourse as pseudo summaries, we extract questions following two rules: (1) utterances that begin with WH-words (e.g. "what"), (2) utterances that end with a question mark. We call this Rule-based Data Augmentation (RBDA). We first pre-train our DDAMS on the two types of pseudo data, which are constructed based on DDADA and RBDA separately. And then fine-tune it on the meeting dataset. The results are shown in Table 5. We can see that DDADA is better than RBDA, demonstrating the effectiveness of dialogue discourse. Besides, pretraining on pseudo-summarization data constructed based on RBDA still achieves a better result, which indicates the rationality of our pretraining strategy.

6 Related Work

Meeting Summarization With the abundance of automatic meeting transcripts, meeting summarization attracts more and more attention [Murray *et al.*, 2010; Shang *et al.*, 2018; Zhao *et al.*, 2019]. There are two main paradigms in meeting summarization. In the first paradigm, some works try to incorporate auxiliary information for better modeling meetings. Goo and Chen [2018] incorporated dialogue acts. Li *et al.* [2019] incorporated topics and vision features. Ganesh and Dingliwal [2019] reconstructed the dialogue into a sequence of words based on discourse labels. In the second paradigm, some works used news summarization data to alleviate the insufficient training problem. Zhu *et al.* [2020] used news data to pretrain their model. Ganesh and Dingliwal [2019] used a news summarizer to generate meeting summaries in a zero-shot manner. In this paper, we propose to use dialogue discourse relations as the auxiliary information to model the interaction between utterances and construct a pseudo-summarization corpus, addressing the data insufficiency problem.

	Model	R-1	R-2	R-L
AMI	DDAMS	51.42	20.99	24.89
	+ RBDA	52.94	21.96	25.05
	+ DDADA	**53.15**	**22.32**	**25.67**
ICSI	DDAMS	39.66	10.09	17.53
	+ RBDA	39.42	10.60	18.19
	+ DDADA	**40.41**	**11.02**	**19.18**

Table 5: The results of pre-training DDAMS on different types of pseudo-summarization data.

Discourse-Aware News Summarization Some works applied a constituency-based discourse structure, namely Rhetorical Structure Theory [Mann and Thompson, 1988] to news summarization [Liu and Chen, 2019; Xu *et al.*, 2020]. These works aimed to analyze the core part of a given sentence and extracted sub-sentential discourse units to form the summary. In this paper, we introduce the dependency-based structure, namely dialogue discourse, for exploring the diverse interactive relations between utterances.

7 Conclusion

In this paper, we apply the dialogue discourse to model the diverse interactions between utterances and present a Dialogue Discourse-Aware Meeting Summarizer (DDAMS). We design a meeting graph to facilitate the information flow and demonstrate that incorporating dialogue discourse is effective for this task. Moreover, we propose a Dialogue Discourse-Aware Data Augmentation (DDADA) strategy to alleviate the insufficient training problem. We build a pseudo-summarization corpus by utilizing the *QA* discourse relation. Experiments on AMI and ICSI datasets show that our model achieves new SOTA performances.

Acknowledgements

This work is supported by the National Key R&D Program of China via grant 2018YFB1005103 and National Natural Science Foundation of China (NSFC) via grant 61906053 and 61976073. We thank all the anonymous reviewers for their insightful comments. We also thank Libo Qin, Yibo Sun and Jiaqi Li for helpful discussion.

References

[Asher *et al.*, 2016] Nicholas Asher, Julie Hunter, Mathieu Morey, Benamara Farah, and Stergos Afantenos. Discourse structure and dialogue acts in multiparty dialogue: the stac corpus. In *LREC*, 2016.

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

[Beck *et al.*, 2018] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. In *ACL*, July 2018.

[Carletta *et al.*, 2005] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*. Springer, 2005.

[Ganesh and Dingliwal, 2019] Prakhar Ganesh and Saket Dingliwal. Restructuring Conversations using Discourse Relations for Zero-shot Abstractive Dialogue Summarization. *arXiv.org*, 2019.

[Goo and Chen, 2018] Chih-Wen Goo and Yun-Nung Chen. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In *SLT*. IEEE, 2018.

[Gross *et al.*, 2013] Jonathan L Gross, Jay Yellen, and Ping Zhang. *Handbook of graph theory*. Chapman and Hall/CRC, 2013.

[Gurevych and Strube, 2004] Iryna Gurevych and Michael Strube. Semantic similarity applied to spoken dialogue summarization. In *COLING*, 2004.

[Janin *et al.*, 2003] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. The icsi meeting corpus. In *ICASSP*. IEEE, 2003.

[Li *et al.*, 2019] Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. Keep meeting summaries on topic: Abstractive multimodal meeting summarization. In *ACL*, July 2019.

[Li *et al.*, 2021] Jiaqi Li, Ming Liu, Zihao Zheng, Heng Zhang, Bing Qin, Min-Yen Kan, and Ting Liu. Dadgraph: A discourse-aware dialogue graph neural network for multiparty dialogue machine reading comprehension. *ArXiv*, 2021.

[Lin, 2004] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, Barcelona, Spain, July 2004. ACL.

[Liu and Chen, 2019] Zhengyuan Liu and Nancy Chen. Exploiting discourse-level segmentation for extractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019.

[Luong *et al.*, 2015] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[Mann and Thompson, 1988] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3), 1988.

[Marcheggiani and Titov, 2017] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP*, 2017.

[Mihalcea and Tarau, 2004] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *EMNLP*, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[Murray *et al.*, 2010] Gabriel Murray, Giuseppe Carenini, and Raymond Ng. Generating and validating abstracts of meeting conversations: a user study. In *INLG*, 2010.

[Nallapati *et al.*, 2017] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, 2017.

[Narayan *et al.*, 2020] Shashi Narayan, Gonçalo Simoes, Ji Ma, Hannah Craighead, and Ryan Mcdonald. Qurious: Question generation pretraining for text generation. *arXiv preprint*, 2020.

[Qin *et al.*, 2020] Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. AGIF: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1807–1816. Association for Computational Linguistics, 2020.

[Sacks *et al.*, 1978] Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. A simplest systematics for the organization of turn taking for conversation. In *Studies in the organization of conversational interaction*. Elsevier, 1978.

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 2018.

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

[Serban *et al.*, 2016] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, 2016.

[Shang *et al.*, 2018] Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *ACL*, 2018.

[Shi and Huang, 2019] Zhouxing Shi and Minlie Huang. A deep sequential model for discourse parsing on multi-party dialogues. In *AAAI*, volume 33, 2019.

[Stone *et al.*, 2013] Matthew Stone, Una Stojnic, and Ernest Lepore. Situated utterances and discourse relations. In *IWCS – Short Papers*, March 2013.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.

[Wang *et al.*, 2020] Tianyi Wang, Yating Zhang, Xiaozhong Liu, Changlong Sun, and Qiong Zhang. Masking orchestration: Multi-task pretraining for multi-role dialogue representation learning. 2020.

[Xu *et al.*, 2020] Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. Discourse-aware neural extractive text summarization. In *ACL*, 2020.

[Zhao *et al.*, 2019] Zhou Zhao, Haojie Pan, Changjie Fan, Yan Liu, Linlin Li, Min Yang, and Deng Cai. Abstractive meeting summarization via hierarchical adaptive segmental network learning. In *The World Wide Web Conference*, 2019.

[Zhu *et al.*, 2020] Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Findings of EMNLP*, 2020.