View Dialogue in 2D: A Two-stream Model in Time-speaker Perspective for Dialogue Summarization and Beyond

Keli Xie, Dongchen He, Jiaxin Zhuang, Siyuan Lu, Zhongfeng Wang*
School of Electronic Science and Engineering, Nanjing University, Nanjing, China
{klxie, dc.he, kkchong, sylu}@smail.nju.edu.cn, zfwang@nju.edu.cn

Abstract

Existing works on dialogue summarization often follow the common practice in document summarization and view the dialogue, which comprises utterances of different speakers, as a single utterance stream ordered by time. However, this single-stream approach without specific attention to the speaker-centered points has limitations in fully understanding the dialogue. To better capture the dialogue information, we propose a 2D view of dialogue based on a time-speaker perspective, where the time and speaker streams of dialogue can be obtained as strengthened input. Based on this 2D view, we present an effective two-stream model called ATM to combine the two streams. Extensive experiments on various summarization datasets demonstrate that ATM significantly surpasses other models regarding diverse metrics and beats the state-of-the-art models on the QMSum dataset in ROUGE scores. Besides, ATM achieves great improvements in summary faithfulness and human evaluation. Moreover, results on machine reading comprehension datasets show the generalization ability of the proposed methods and shed light on other dialogue-based tasks. Our code will be publicly available online.[1]

1 Introduction

Dialogue summarization is a task aiming to generate a succinct and coherent summary of the given dialogue, which has been explored in many applications, such as automatic meeting summarization, and drawn the attention of many researchers.

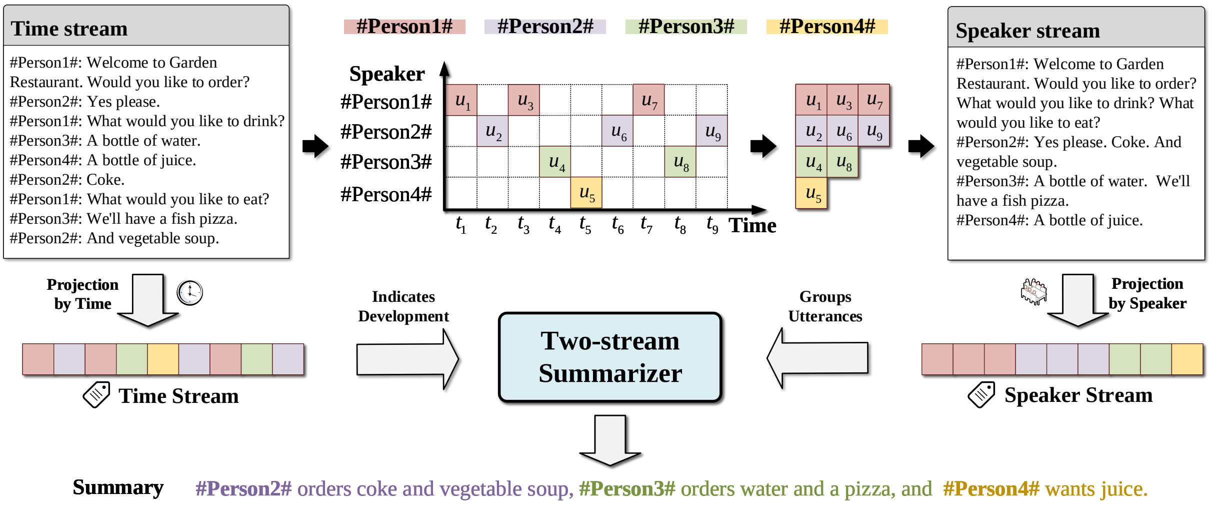
With the development of Transformer-based pretrained models (Vaswani et al., 2017; Lewis et al., 2020; Zhang et al., 2020a), remarkable progress has been made in text summarization, especially in document summarization (El-Kassas et al., 2021). However, dialogue summarization is still quite challenging partly due to the structural characteristics of dialogue text (Feng et al., 2021a). A major difference is that a document is often organized with a unified narrative perspective, while a dialogue includes many speakers that bring diversity and switches of narrative perspectives (Kryscinski et al., 2021). Moreover, the information from different speakers is scattered (Liu et al., 2021), which poses a challenge to the summarizer as the dialogue summary is often speaker-centered that focuses on the speaker's actions and opinions (Xu and Lapata, 2021; Zhong et al., 2021b).

Current works on dialogue summarization usually view the dialogue as a single utterance stream ordered by time (Zhong et al., 2021a), like prior studies on document summarization do (Lin and Ng, 2019). This single-stream approach, however, has limitations in fully using the information about dialogue (Lei et al., 2021b). Firstly, the utterances of different speakers are interlaced in the single stream, which may weaken the semantic continuity from each speaker's perspective. Besides, to obtain a concise summary instead of a laundry list, we need to both notice the development in time order and sum up information about each speaker (Zhao et al., 2022). The single-stream approach may not be enough since it mainly focuses on the former. Furthermore, the summarizer needs to deal with the frequent switches of multiple narrative perspectives in the single stream. The generated summaries are consequently often accompanied by unfaithfulness problems, such as coreference error and missing information (Maynez et al., 2020), which hinder the practical applications of dialogue summarizers.

To tackle these challenges, we propose a *2D view* that restructures the dialogue text to make the most of dialogue information. Inspired by how humans summarize, we arrange the dialogue in a time-speaker view as Figure 1 shows, where time and speaker streams can be obtained by projecting the dialogue. Intuitively, the two streams are two directions for humans to summarize a dialogue.

[*] Corresponding author
[1] https://github.com/shakeley/View2dSum

Proceedings of the 29th International Conference on Computational Linguistics, pages 6075–6088
October 12–17, 2022.



Time stream

#Person1#: Welcome to Garden Restaurant. Would you like to order?
#Person2#: Yes please.
#Person1#: What would you like to drink?
#Person3#: A bottle of water.
#Person4#: A bottle of juice.
#Person2#: Coke.
#Person1#: What would you like to eat?
#Person3#: We'll have a fish pizza.
#Person2#: And vegetable soup.

Speaker stream

#Person1#: Welcome to Garden Restaurant. Would you like to order? What would you like to drink? What would you like to eat?
#Person2#: Yes please. Coke. And vegetable soup.
#Person3#: A bottle of water. We'll have a fish pizza.
#Person4#: A bottle of juice.

Summary: #Person2# orders coke and vegetable soup, #Person3# orders water and a pizza, and #Person4# wants juice.

Figure 1: Overview of our methods. u_i indicates the i-th utterance in the dialogue. The two streams can be seen as projections onto time and speaker dimensions, respectively, which complement each other. The speaker stream can help align with the summary as the dialogue summary is often speaker-centered (shown in different colors).

The time stream is the same as the dialogue in the 1D view that helps understand the development of the dialogue. The speaker stream comprises utterances grouped by speaker, which is beneficial for summing up information about different speakers and improving the faithfulness as it reduces the switches of narrators. Besides, the speaker stream is *resource-friendly*, for it is automatically generated without any resource-consuming annotations by humans or big models (Feng et al., 2021c).

The two streams focus on two complementary aspects of dialogue. To combine these two streams, we present a *two-stream model* called ATM based on Transformer. The ATM encoder consists of a trunk-branch network to catch the salient commonality and individuality of the two streams. Then we leverage two cross-attention modules in each ATM decoder layer to capture the information of both time and speaker streams. The approach can be easily applied to other Transformer-based models.

Extensive experimental results on three dialogue summarization datasets show that ATM significantly surpasses other baseline models by a large margin regarding various automatic metrics and achieves new state-of-the-art results on QMSum dataset, to the best of our knowledge. ATM also mitigates unfaithfulness problems and achieves higher scores in our human evaluation.

Moreover, to test the generalization ability of the proposed methods, we conduct experiments on two Machine Reading Comprehension (MRC) datasets. The results demonstrate that ATM also outperforms the single time-stream model in MRC task.

Our main contributions are three-fold. 1) We propose a novel 2D view for better representing dialogue. 2) A two-stream model is presented for dialogue called ATM to make the most of dialogue information. 3) We conduct extensive experiments to demonstrate the effectiveness and insights of the proposed methods in two dialogue-based tasks.

2 Related Work

Abstractive Dialogue Summarization has attracted much attention recently since abstractive methods can produce more coherent and Readabie summaries than extractive methods. In early stage, Banerjee et al. (2015) utilize the dependency graph. Oya et al. (2014) and Singla et al. (2017) explore template-based methods. With the development of deep learning and publicly available datasets (Zhong et al., 2021b; Chen et al., 2021, 2022), plenty of related works have been conducted. To utilize the interactive characteristic of dialogue, graph-based methods are used in (Shang et al., 2018; Zhao et al., 2020; Feng et al., 2021b). For capturing the acts in dialogue, Goo and Chen (2018) propose a sentence-gated mechanism and Di et al. (2020) use dialogue acts as an interactive pattern. Chen and Yang (2021) incorporate discourse relations. Unsupervised strategies are explored in (Zou et al., 2021; Fu et al., 2021; Zhang et al., 2021). Feng et al. (2021c) and Yuan and Yu (2019) leverage annotators to dig up more information. A specialized pre-training framework is proposed in (Zhong et al., 2021a). As dialogue is composed of multiple turns, HMNet (Zhu et al., 2020), Manakul et al. (2020) and Qi et al. (2021) use a hierarchical network to model multi-level representations of dialogue. Chen and Yang (2020) propose a multi-view approach. A major limitation of these works is that they view the dialogue from a 1D perspective, while we arrange the dialogue in a time-speaker view. They also lack enough attention to the semantic continuity that we focus on.

Speaker-aware Methods for Improving Faithfulness focus on the speakers in the dialogue. The multiple speakers contain helpful information while posing a challenge for the model to generate faithful summaries (Maynez et al., 2020). To relieve the confusion of personal pronouns, Lei et al. (2021a) propose a from-coarse-to-fine procedure. FinDS (Lei et al., 2021b) utilizes finer-grain semantic structures to clarify the speaker relationships. Lee et al. (2021) propose a self-supervised strategy to do post-correction for speakers. Zhao et al. (2022) leverage a speaker-aware structure to model the interaction process in dialogue.

Two-stream Architectures have been utilized in several areas, including Computer Vision (Simonyan and Zisserman, 2014; Chen et al., 2018; Sevilla-Lara et al., 2018; Kwon, 2021), Natural Language Processing including XLNet (Yang et al., 2019) and ERNIE-Gram (Xiao et al., 2021), and Multimodal applications including LXMERT (Tan and Bansal, 2019), ViLBERT (Lu et al., 2019) and ERNIE-ViL (Yu et al., 2021). Other works include adding extra modules for specific purposes, such as adaptive computing (Wang et al., 2022; Xie et al., 2021). Our originality lies in applying the two-stream idea to dialogue text based on the proposed time-speaker view.

3 Methodology

In this section, we first introduce the time-speaker view of dialogue and present the problem formulation of the two-stream summarization based on this view. Then details of ATM are presented.

3.1 Time-speaker View of Dialogue

Motivated by the significance of speakers in dialogue, we propose a novel time-speaker view as shown in Figure 1 to better represent the dialogue. Unlike the traditional 1D view with only the time stream of dialogue, the 2D view highlights the speaker of each utterance. From this viewpoint, the time stream can be seen as the projection of dialogue onto the time dimension. Meanwhile, some information from the speaker dimension is missing due to projection. The common practice is to add the corresponding speaker in the front of each utterance as a supplement. However, this approach still leaves the problems of scattered information about speakers, frequent switches of narrative perspectives, and unfaithfulness in generated summaries.

Algorithm 1 Speaker Stream $\mathbf{x}^s$

```
Input: $\mathbf{x}^t = \{u_1^t, ..., u_n^t\}$
Output: $\mathbf{x}^s$
  $S,\ T \Leftarrow \{S_1, ..., S_m\},\ \{T_1, ..., T_m\}$
  Initialize $T_j \in T$ with $S_j \in S$
  for $u_i^t = \{s_i^t,\ c_i^t\} \in \mathbf{x}^t$ do
    for $S_j \in S$ do
      if $s_i^t = S_j$ then
        $T_j \Leftarrow concat(T_j,\ c_i^t)$
      end if
    end for
  end for
  return $\mathbf{x}^s \Leftarrow concat(T_1, ..., T_m)$
```

To address these issues, we obtain the speaker stream from the projection onto the speaker dimension. The purpose of summarization is to get the main points that often focus on the actions and opinions of speakers when it comes to dialogue. The speaker stream gathers each speaker's utterances, thus providing a simple way for the model to capture the information about speakers. Additionally, the speaker stream can help the model align with the summary as the dialogue summary is often speaker-centered. Our method of generating the speaker stream serves as a baseline method in the two-stream scheme, which is easy to follow and generalize to other tasks. Researchers can also use other methods to generate the speaker stream, which can be explored in future work.

In general, the two streams complement each other as the time stream helps understand the development of dialogue and the speaker stream is useful for catching speaker-centered information and improving faithfulness of the generated text. Therefore, we combine the time and speaker streams in a two-stream model described in Section 3.3.

3.2 Problem Formulation

Given a dialogue source example $\mathbf{d}$ that comprises n utterances, the time stream can be denoted as $\mathbf{x}^t = \{u_1^t, ..., u_n^t\}$, where u_i^t indicates the i-th utterance of $\mathbf{d}$ in time order. Each u_i^t consists of

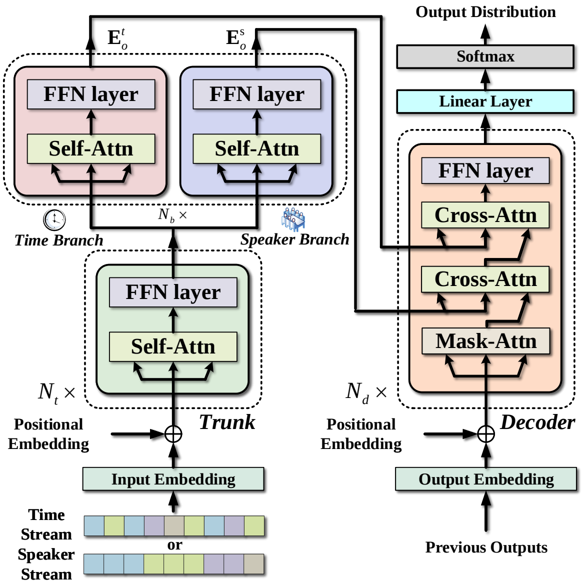


Figure 2: Main architecture of ATM with one branch layer. Colors in two streams indicate different speakers.

speaker s_i^t and content c_i^t. The speaker stream $\mathbf{x}^s$ can be obtained with Algorithm 1, where S contains m speakers in the order they appear in $\mathbf{d}$ and $concat(\cdot)$ denotes string concatenation function.

Then $\mathbf{x}^t$ and $\mathbf{x}^s$ are sent to a two-stream model to generate summaries. The training objective is to maximize the conditional likelihood of the outputs $\mathbf{y}$, which can be represented as:

$$\max_{\theta} \sum_{k=1}^{|\mathcal{D}|} \log p_\theta(\mathbf{y}_k|\mathbf{x}_k^t, \mathbf{x}_k^s), \tag{1}$$

where θ denotes the model parameters and $\mathcal{D}$ indicates the training examples. Teacher-forcing strategy(Williams and Zipser, 1989) is used in training.

3.3 ATM Architecture

Figure 2 illustrates the main architecture of the proposed two-stream model ATM inspired by Ding and Tao (2018). ATM's backbone is based on Transformer (Vaswani et al., 2017). To combine the two streams, we introduce a Two-stream Encoder and a Two-stream Decoder.

3.3.1 Two-stream Encoder

The two-stream encoder is a trunk-branch network with a trunk depth N_t and a branch depth N_b. The original depth of encoder $N_e = N_t + N_b$. The encoder mainly includes self-attention layers, layer normalization modules, and feed-forward layers. The time stream $\mathbf{x}^t$ and speaker stream $\mathbf{x}^s$ are sent to the encoder to get the contextual representations of each stream.

Specifically, $\mathbf{x}^t$ and $\mathbf{x}^s$ go through the embedding layers and the same N_t trunk layers for obtaining basic representations $\mathbf{E}'^t$ and $\mathbf{E}'^s$. Then we adopt N_b branch layers to further encode each stream separately and get the encoder outputs $\mathbf{E}_o^t$ and $\mathbf{E}_o^s$. The encoding process can be denoted as:

$$\begin{aligned} \mathbf{e}^t; \mathbf{e}^s &= Embedding(\mathbf{x}^t; \mathbf{x}^s) \\ \mathbf{E}'^t; \mathbf{E}'^s &= Trunk(\mathbf{e}^t; \mathbf{e}^s) \\ \mathbf{E}_o^t &= TimeBranch(\mathbf{E}'^t) \\ \mathbf{E}_o^s &= SpeakerBranch(\mathbf{E}'^s). \end{aligned} \tag{2}$$

Using this trunk-branch structure, we expect the encoder to capture the salient commonality and individuality of the two streams with trunk and branch layers, respectively. Also, we can save additional parameters compared with using two individual encoders by this approach. We set $N_b = 1$ and $N_e = 12$ while adopting the pre-trained parameters of BART-large (Lewis et al., 2020) in both trunk and branch layers.

3.3.2 Two-stream Decoder

The two-stream decoder inherits the structure of Transformer decoder with $N_d = 12$. The critical difference is that we adopt two cross-attention modules to capture information from both the time and speaker streams.

As Figure 2 shows, the former cross-attention module attends to the encoder output of speaker stream $\mathbf{E}_o^s$ in each decoder layer. Next, the encoder output of time stream $\mathbf{E}_o^t$ is utilized in the latter cross-attention module. In this way, the decoder will first have general impressions of each speaker and focus on the details in time order. This arrangement is inspired by reading novels. If we first get general information like each character's experience and then read the story's development in time order, our understanding will be more comprehensive and faithful. We believe that this approach is also helpful in understanding the dialogue. Note that pre-trained parameters of BART are employed in the cross-attention modules for $\mathbf{E}_o^t$, while those for $\mathbf{E}_o^s$ are randomly initialized. The impact of different ways to use pre-trained parameters will be discussed in Section 5.3.

Dataset	Task	Domain	Size			# *Avg* Tokens		# Speakers
			Train	*Valid*	*Test*	*Src*	*Ref*	*Avg / Max*
QMSum	Summ	Meetings	1,257	272	279	8,263	70	9.2 / 105.0
SummScreen	Summ	TV series	18,915	1,795	1,793	6,613	337	25.5 / 92.0
DialogSum	Summ	Daily life	12,460	500	500	131	24	2.0 / 7.0
QAConv	MRC	Conversations	27,287	3,414	3,505	233	3	3.2 / 14.0
Molweni	MRC	Chat	8,771	883	100	104	4	3.5 / 9.0

Table 1: Datasets evaluated from various domains. Summ indicates summarization. *Avg* denotes average number.

4 Experiment

4.1 Datasets

We conduct extensive experiments on datasets from various domains as Table 1 shows.

QMSum (Zhong et al., 2021b) is a query-based summarization dataset from meetings including AMI (Carletta et al., 2005) and ICSI (Janin et al., 2003). We use the version with gold spans selected by experts.

SummScreen (Chen et al., 2022) is a summarization dataset of TV series transcripts. We use its TMS version for it provides the official recaps.

DialogSum (Chen et al., 2021) is a dialogue summarization dataset from real-life scenarios.

QAConv (Wu et al., 2021) is an MRC dataset that uses conversations as a knowledge source and includes extractive and abstractive answer types.

Molweni (Li et al., 2020) is an MRC dataset that derives from Ubuntu Chat Corpus (Lowe et al., 2015) which consists of multi-party dialogues.

4.2 Baseline Models

Transformer (Vaswani et al., 2017) is a seq-to-seq model relying on an attention mechanism.

Longformer (Beltagy et al., 2020) is a scalable Transformer for processing long documents.

UNILM (Dong et al., 2019) is a unified pre-trained language model. UNILM-CP is further trained on MediaSum (Zhu et al., 2021) and OpenSubtitles (Lison and Tiedemann, 2016) corpora.

HMNet (Zhu et al., 2020) is a hierarchical network for abstractive dialogue summarization with cross-domain pre-training.

BART (Lewis et al., 2020) is an effective pre-trained model with a Transformer architecture for various tasks including summarization.

SUMM^N (Zhang et al., 2022) is a multi-stage network using BART for long text summarization.

4.3 Evaluation Metrics

Various metrics are adopted for a rigorous evaluation, including n-gram overlap, model-based, and faithfulness-aware methods.

ROUGE (Lin, 2004) is a widely used automatic metric for summarization, based on lexical overlaps between a reference and the generated text.

BERTScore (Zhang et al., 2020b) is a metric for text generation based on semantic similarity.

BARTScore (Yuan et al., 2021) is a new evaluation metric, which can evaluate generated text as text generation from different perspectives.

FactCC (Kryscinski et al., 2020) is a factual consistency checking model for text summarization.

SUMMAC (Laban et al., 2022) is a novel NLI-based (Natural Language Inference) model for summary inconsistency detection. We use the SUMMAC$_{Conv}$ version for our evaluation as it performs well in the original paper.

MRC Evaluation Metrics For MRC datasets, we report exact match (EM), F1 scores, and FZ-R scores following the common practice. The EM means that predicted answers must be the same as the ground truth. The F1 score is calculated by tokens overlapping. We also present the FZ-R scores, which use the Levenshtein distance to calculate the differences between two sequences.

4.4 Implementation Details

Training & Generation We use the `fairseq`[2] (Ott et al., 2019) implementation for BART-large. The experiments are done on a single NVIDIA RTX 3090 GPU with a 24GB memory. The total number of parameters of ATM is 469M and that of BART-large is 406M. The max number of input tokens is set to 2048 by default. The dropout rate is 0.1. An early stop patience of 3 is used in our

[2] https://github.com/pytorch/fairseq

Model	QMSum			SummScreen			DialogSum		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Transformer	-	-	-	-	-	-	35.91	8.74	33.50
Longformer	31.60	7.80	20.50	42.90	11.90	41.60	-	-	-
UniLM-base	29.14	6.25	25.46	43.42	9.62	41.19	-	-	-
UniLM-CP	29.19	6.73	25.52	44.07	9.96	41.73	-	-	-
HMNet	36.06	11.36	31.27	-	-	-	-	-	-
Summ^N	40.20	15.32	35.62	44.64	11.87	42.53	-	-	-
BART	37.02	14.23	27.49	43.59	10.37	41.43	46.01	20.78	41.06
ATM (Ours)	**40.43**	**16.27**	**36.08**	**44.69**	**12.82**	**43.11**	**46.49**	**21.12**	**41.56**

Table 2: Main results on QMSum, SummScreen, and DialogSum summarization datasets. R is short for ROUGE. The results of BART are from our tests and other results are from the corresponding papers of models or datasets.

Model	QMSum		SummScreen		DialogSum	
	ES	AS	ES	AS	ES	AS
BART	86.11	-3.68	84.01	-3.61	91.53	**-2.08**
ATM	**87.48**	**-3.14**	**84.25**	**-3.48**	**91.84**	-2.09

Table 3: ES (BERTScore) and AS (BARTScore) scores.

experiments. We do grid searching for some hyperparameters, such as learning rate, warmup step, and gradient accumulation step for BART, making our best efforts for a fair comparison. The detailed settings are included in Appendix A.

Evaluation We adopt `files2rouge`[3] library for ROUGE scores. For other metrics, we use the officially released codes described in Appendix B.

MRC Setting BART can be seen as a free-form model to generate predicted answers given the dialogues and questions. The question is added at the front of each dialogue separated by a special token.

5 Results and Analysis

5.1 Main Results on Summarization

Effectiveness of ATM Table 2 shows the main results of ROUGE scores. The proposed ATM achieves the best performances among other baselines on various datasets. Compared with BART, the original single-stream model, ATM improves the scores by a large margin, which shows the effectiveness of the proposed methods.

Concretely, ATM achieves new state-of-the-art results on QMSum, to the best of our knowledge. The improvements are 3.41, 2.04, and 8.59 for ROUGE-1, ROUGE-2, and ROUGE-L, respectively. As for SummScreen, ATM boosts by 1.10, 2.45, and 1.68 for ROUGE-1, ROUGE-2, and ROUGE-L compared to BART. For DialogSum, ATM brings improvements as well.

ATM also achieves broadly better results than BART in two model-based metrics, BERTScore and BARTScore, as Table 3 shows[4]. The above results of various metrics show the effectiveness of the proposed methods.

Advantages over the Strong Baseline Summ^N (Zhang et al., 2022) uses multiple BARTs for multi-stage summarization. Table 2 shows that ATM achieves better results than this powerful baseline. Besides, Summ^N contains at least 812M parameters compared to 469M of ATM and brings huge computation costs. Another difference is that BART further trained on CNN/DM (Hermann et al., 2015) dataset is used in Summ^N, while we choose the original checkpoint for a more transparent comparison of the methods themselves.

Dramatic Boost on Query-based Dataset Among these experimental datasets, the results of QMSum show the most considerable improvement. We attribute this to the match between the two-stream model and the characteristics of QMSum. The examples in QMSum include many questions on specific speakers, such as `What does the Manager say about the plan`. This characteristic echoes the speaker-centered feature of the dialogue summary. It also sets a higher bar for the summary quality that we can tell from the relatively low ROUGE scores. Hence ATM may benefit from the speaker stream that helps focus on the utterances of certain speakers.

5.2 Faithfulness Evaluation

Besides ROUGE scores, the improvement of faithfulness is another critical topic to help summariza-

[3] `https://github.com/pltrdy/files2rouge`

[4] The comparison is mainly between ATM and BART since ATM is initialized with BART.

Model	QMSum		SummScreen		DialogSum	
	FC	SC	FC	SC	FC	SC
BART	78.23	43.12	**96.13**	20.01	88.18	20.12
ATM	**80.02**	**48.13**	96.01	**20.11**	**89.04**	**20.32**

Table 4: Results of faithfulness evaluation. FC and SC indicate **F**actC**C** and **S**UMMA**C**, respectively.

tion models be applied in practice. We conduct faithfulness evaluation across classifier-based and NLI-based methods as shown in Table 4. By and large, ATM achieves better results on selected faithfulness metrics compared with BART. The results demonstrate the effectiveness of ATM in improving the faithfulness of generated summaries as it reduces narrator switches. The concrete case study is presented in Appendix C for illustration.

5.3 Ablation Study

To investigate the effect of the proposed methods, we make ablation experiments on QMSum from the perspectives of model input and structure.

5.3.1 Input-wise Ablations

Effectiveness of Using Two Streams for ATM For ATM, feeding a single stream to both time and speaker branches leads to much lower scores than using two streams, as Table 5 shows. This observation indicates that the two streams bring additional improvements. We attribute this to their complementary feature as mentioned in Section 3.1.

What if Only Using Speaker Stream As Table 5 shows, we feed the speaker stream alone to ATM and BART. Although the results are not as good as those of using single time stream input, the model still achieves comparable performance. This finding indicates that the model can indeed utilize the semantic information of the speaker stream. Meanwhile, the comparative advantage of the time stream may partly come from the speaker added in front of each utterance as a soft prompt for information in the speaker dimension, while it is hard for the single speaker stream to do so.

Can We Just Concat Two Streams Admittedly, ATM incorporates extra modules to implement a two-stream model. Hence we use BART to process a concatenated input of the two streams as a simpler model. As shown in Table 5, the performance is even worse than that of using the single time stream. We attribute this to the confusion caused by the concatenated input, i.e., it is hard for the model to understand the two-stream input organized in different ways with the same structure. Besides, ATM processing single stream still achieves higher scores than BART. The above results demonstrate the necessity and effectiveness of ATM.

Methods	R-1	R-2	R-L
Input-wise			
ATM			
- time stream	38.68	14.53	34.02
- speaker stream	37.86	13.93	33.95
- two streams	**40.43**	**16.27**	**36.08**
BART			
- time stream	37.02	14.23	27.49
- speaker stream	36.53	13.78	26.99
- *concat* two streams	36.77	13.95	27.38
Structure-wise			
# Branch layers in ATM			
- with 0 branch layer	37.88	14.36	34.19
- with 1 branch layer	**40.43**	**16.27**	**36.08**
- with 2 branch layers	39.73	15.61	35.19
Use of pre-trained param			
- not ptr. speaker branch	39.45	14.82	34.73
- ptr. speaker cross-attn	39.56	15.36	34.99
Order of cross-attn			
- t.s. cross-attn	39.65	14.91	35.18

Table 5: Ablations on QMSum. time/speaker stream for ATM denotes feeding the same stream to the time and speaker branches. *concat* represents concatenate. ptr. indicates using pre-trained parameters. t.s. means the decoder attends to time stream first then speaker stream.

5.3.2 Structure-wise Ablations

A Balanced Trunk-branch is Needed in Encoder We test with different numbers of branch layers N_b while the encoder depth N_e remains the same. As shown in Table 5, the model sharing all the encoder layers achieves the lowest scores. As N_b grows, the ROUGE scores first increase and then decrease. This observation indicates that the encoder needs a balanced trunk-branch structure to combine the commonality and individuality of the two streams, thus achieving stronger performance.

Effect of Pre-trained Parameters We can tell from Table 5 that both using pre-trained parameters in cross-attention for speaker stream and randomly initializing speaker branch achieve lower scores than ATM. We attribute this to the different features of the encoder and decoder. For the encoder, the purpose of encoding for both two streams is similar: to get contextual representations. Hence

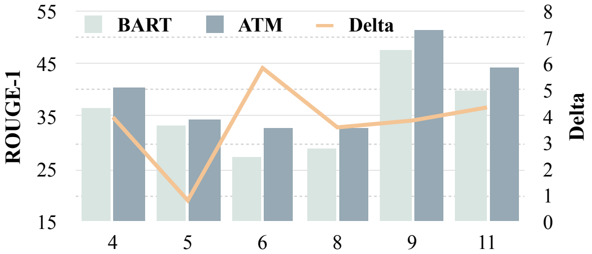


Figure 3: Impact of number of speakers. Delta indicates the performance gap between BART and ATM.

the pre-trained parameters are compatible with both branches. For the decoder, the pre-trained parameters are obtained based on an autoregressive decoding process following the time order. There may be a mismatch between pre-trained parameters and cross-attention modules for the speaker stream as it is not in time order.

Order of Two Cross-attention Modules We change the order of cross-attention modules from speaker-time to time-speaker and get worse performance. This gap may also result from the autoregressive feature of decoding. There is a chance that the model fails to see the wood for the trees if we first attend to the time stream.

5.4 Impact of Number of Speakers

We report experimental results on QMSum dataset in Figure 3 to examine how the number of speakers affects the model performance. Interestingly, ATM generally achieves more significant improvements with more speakers. The fluctuation may come from the number and complexity of evaluation examples. This finding is in line with one mentioned feature of dialogue, i.e., the multiple speakers pose challenges for summarizers. With more speakers, the speaker information is more scattered. Due to the speaker stream that groups utterances by different speakers, ATM has an advantage in gathering speaker information and achieves a greater boost.

5.5 Ability of Abstraction

To compare the model ability of abstraction, we calculate the percentage of novel n-grams (i.e., n-grams that do not appear in the source text) in the generated summaries. As Figure 4 shows, the percentage of ATM is higher than BART, which indicates that ATM is good at generating abstractive summaries. We attribute this to the speaker stream that provides a direction for abstraction precisely.

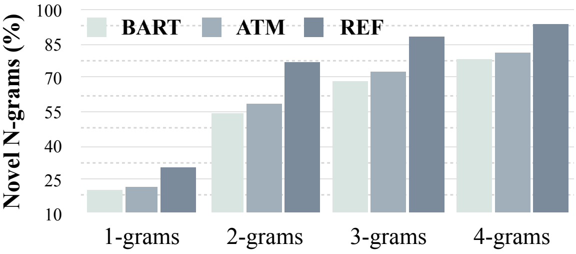


Figure 4: Percentage of novel n-grams in the generated summaries and **REF**erence summary on DialogSum.

Model	Complt.	Readabi.	Faith.
BART	3.51	4.14	3.21
ATM	**3.78**	**4.52**	**3.54**

Table 6: Human evaluation results. Complt., Readabi., and Faith. denote completeness, readability and faithfulness, respectively.

5.6 Human Evaluation

We perform a human evaluation by three human evaluators to assess the completeness, readability and faithfulness of the generated summaries. The evaluators are asked to rate the different summaries on a Likert Scale from 1 to 5. Completeness measures how well the summary includes key information. Readability measures how well the summary is coherent and concise. Faithfulness measures how well the summary includes reliable information. We take a random 10% sample of the DialogSum test set. The two generated summaries are randomly ordered for each dialogue for reducing bias. The evaluators read both the dialogue script and the corresponding summaries to score from 1 to 5 (higher is better). As Table 6 shows, ATM achieves higher scores among three metrics than BART, which indicate the advantage of ATM. Besides, we compute the Fleiss's Kappa scores (k) (Fleiss, 1971) to assess the agreement among the raters. The scores all lie in $(0.6 \leq k \leq 0.8)$, which show substantial agreement among the evaluators.

5.7 Beyond Dialogue Summarization

To show the potential of ATM as a universal approach to improving the performance on dialogue-based tasks, we conduct experiments on two MRC datasets in dialogue domain. As shown in Table 7, compared to BART, ATM significantly boosts the EM to 71.57 by 2.12 on QAConv dataset. The improvements in F1 and FZ-R are remarkable as well. For Molweni dataset, ATM achieves about 1.6 higher scores than BART for all three metrics.

Model	QAConv			Molweni		
	EM	F1	FZ-R	EM	F1	FZ-R
BERT-base†	66.4	76.3	81.3	/	58.0	/
BERT-large†	72.9	81.7	85.6	/	65.5	/
T5-base†	71.2	80.9	84.7	/	/	/
T5-large†	73.5	83.0	86.6	/	/	/
BART	69.4	78.7	83.4	45.1	65.2	73.3
ATM	71.6	79.6	84.3	46.8	66.8	74.8

Table 7: MRC evaluation on QAConv and Molweni. The metrics are described in Section 4.3. †The results on QAConv of BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020) come from Wu et al. (2021). BERT and T5 are further trained on SQuAD (Rajpurkar et al., 2016) and UnifiedQA (Khashabi et al., 2020), respectively. The BERT results on Molweni are from Li et al. (2020).

The results demonstrate the generalization ability of the proposed methods on MRC and shine a light on other dialogue-based tasks. Meanwhile, the score gap between BART and other models may result from the different architectures and usages of additional in-domain data as shown in Table 7.

6 Conclusion

To make the most of dialogue information, we propose a novel 2D view highlighting the speakers based on time-speaker space, which provides a new inspiring perspective on modelling dialogue. Then a simple and effective two-stream summarization model ATM is presented to utilize the information from both the time and speaker streams obtained from this view. Empirical results demonstrate that the proposed methods surpass other models on three summarization datasets regarding various metrics, faithfulness and human evaluation. We also show the significant improvements on MRC, a representative of other dialogue-based tasks.

This work leaves several open directions that can be explored, including 1) applying the proposed methods to other models and tasks, 2) exploring the 2D view from other directions, and 3) establishing a general framework for dialogue-based tasks.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. Thanks to DME for the support during this work. This work was supported in part by the National Natural Science Foundation of China under Grant 62174084, 62104097, in part by the High-Level Personnel Project of Jiangsu Province under Grant JSSCBS20210034, and in part by the Key Research Plan of Jiangsu Province of China under Grant BE2019003-4.

References

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Abstractive meeting summarization using dependency graph fusion. In *Proceedings of the 24th international conference on world wide web*, pages 5–6.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

Di Chen, Shanshan Zhang, Wanli Ouyang, Jian Yang, and Ying Tai. 2018. Person search via a mask-guided two-stream cnn model. In *Proceedings of the european conference on computer vision (ECCV)*, pages 734–750.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4106–4118, Online. Association for Computational Linguistics.

Jiaao Chen and Diyi Yang. 2021. Structure-aware abstractive conversation summarization via discourse and action graphs. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1380–1391, Online. Association for Computational Linguistics.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiasheng Di, Xiao Wei, and Zhenyu Zhang. 2020. How to interact and change? abstractive dialogue summarization with dialogue act weight and topic change info. In *Knowledge Science, Engineering and Management*, pages 238–249, Cham. Springer International Publishing.

Changxing Ding and Dacheng Tao. 2018. Trunk-branch ensemble convolutional neural networks for video-based face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):1002–1014.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021a. A survey on dialogue summarization: Recent advances and new frontiers. *CoRR*, abs/2107.03175.

Xiachong Feng, Xiaocheng Feng, Bing Qin, and Xinwei Geng. 2021b. Dialogue discourse-aware graph model and data augmentation for meeting summarization. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3808–3814. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021c. Language model as an annotator: Exploring DialoGPT for dialogue summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1479–1491, Online. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Xiyan Fu, Yating Zhang, Tianyi Wang, Xiaozhong Liu, Changlong Sun, and Zhenglu Yang. 2021. RepSum: Unsupervised dialogue summarization based on replacement strategy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6042–6051, Online. Association for Computational Linguistics.

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.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 2003. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, volume 1, pages I–I.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Wojciech Kryscinski, Nazneen Fatema Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir R. Radev. 2021. Booksum: A collection of datasets for long-form narrative summarization. *CoRR*, abs/2105.08209.

Soonil Kwon. 2021. Optimal feature selection based speech emotion recognition using two-stream deep convolutional neural network. *International Journal of Intelligent Systems*, 36(9):5116–5135.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Dongyub Lee, Jungwoo Lim, Taesun Whang, Chanhee Lee, Seungwoo Cho, Mingun Park, and Heuiseok Lim. 2021. Capturing speaker incorrectness: Speaker-focused post-correction for abstractive dialogue summarization. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 65–73, Online and in Dominican Republic. Association for Computational Linguistics.

Yuejie Lei, Yuanmeng Yan, Zhiyuan Zeng, Keqing He, Ximing Zhang, and Weiran Xu. 2021a. Hierarchical speaker-aware sequence-to-sequence model for dialogue summarization. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7823–7827.

Yuejie Lei, Fujia Zheng, Yuanmeng Yan, Keqing He, and Weiran Xu. 2021b. A finer-grain universal dialogue semantic structures based model for abstractive dialogue summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1354–1364, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiaqi Li, Ming Liu, Min-Yen Kan, Zihao Zheng, Zekun Wang, Wenqiang Lei, Ting Liu, and Bing Qin. 2020. Molweni: A challenge multiparty dialogues-based machine reading comprehension dataset with discourse structure. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2642–2652, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Hui Lin and Vincent Ng. 2019. Abstractive summarization: A survey of the state of the art. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9815–9822.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Junpeng Liu, Yanyan Zou, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Caixia Yuan, and Xiaojie Wang. 2021. 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.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic. Association for Computational Linguistics.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Potsawee Manakul, Mark J.F. Gales, and Linlin Wang. 2020. Abstractive Spoken Document Summarization Using Hierarchical Model with Multi-Stage Attention Diversity Optimization. In *Proc. Interspeech 2020*, pages 4248–4252.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. 2014. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 45–53, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.

MengNan Qi, Hao Liu, YuZhuo Fu, and Ting Liu. 2021. Improving abstractive dialogue summarization with hierarchical pretraining and topic segment. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1121–1130, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Laura Sevilla-Lara, Yiyi Liao, Fatma Güney, Varun Jampani, Andreas Geiger, and Michael J Black. 2018. On the integration of optical flow and action recognition. In *German conference on pattern recognition*, pages 281–297. Springer.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 664–674, Melbourne, Australia. Association for Computational Linguistics.

Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199.

Karan Singla, Evgeny Stepanov, Ali Orkan Bayer, Giuseppe Carenini, and Giuseppe Riccardi. 2017. Automatic community creation for abstractive spoken conversations summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 43–47, Copenhagen, Denmark. Association for Computational Linguistics.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. 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 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Meiqi Wang, Liulu He, Jun Lin, and Zhongfeng Wang. 2022. Rethinking adaptive computing: Building a unified model complexity-reduction framework with adversarial robustness. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1803–1810.

Ronald J. Williams and David Zipser. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280.

Chien-Sheng Wu, Andrea Madotto, Wenhao Liu, Pascale Fung, and Caiming Xiong. 2021. Qaconv: Question answering on informative conversations. *arXiv preprint arXiv:2105.06912*.

Dongling Xiao, Yu-Kun Li, Han Zhang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE-gram: Pre-training with explicitly n-gram masked language modeling for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1702–1715, Online. Association for Computational Linguistics.

Keli Xie, Siyuan Lu, Meiqi Wang, and Zhongfeng Wang. 2021. Elbert: Fast albert with confidence-window based early exit. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7713–7717.

Yumo Xu and Mirella Lapata. 2021. Generating query focused summaries from query-free resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6096–6109, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie-vil: Knowledge enhanced vision-language representations through scene graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3208–3216.

Lin Yuan and Zhou Yu. 2019. Abstractive dialog summarization with semantic scaffolds. *arXiv preprint arXiv:1910.00825*.

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 Liu. 2020a. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

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*.

Xuchao Zhang, Bo Zong, Wei Cheng, Jingchao Ni, Yanchi Liu, and Haifeng Chen. 2021. Unsupervised concept representation learning for length-varying text similarity. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5611–5620, Online. Association for Computational Linguistics.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Awadallah, Dragomir Radev, and Rui Zhang. 2022. Summ^n: A multi-stage summarization framework for long input dialogues and documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1592–1604, Dublin, Ireland. Association for Computational Linguistics.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lulu Zhao, Weiran Xu, Chunyun Zhang, and Jun Guo. 2022. Leveraging speaker-aware structure and factual knowledge for faithful dialogue summarization. *Knowledge-Based Systems*, 245:108550.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021a. Dialoglm: Pre-trained model for long dialogue understanding and summarization. *arXiv preprint arXiv:2109.02492*.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021b. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. 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 *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5927–5934, Online. Association for Computational Linguistics.

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 194–203, Online. Association for Computational Linguistics.

Han Zou, Jianfei Yang, and Xiaojian Wu. 2021. Unsupervised energy-based adversarial domain adaptation for cross-domain text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1208–1218, Online. Association for Computational Linguistics.

Dataset	**LR**	**MT**	**GStep**	**WStep**
QMSum	3e-05	2048	4	100
SummScreen	7e-05	1024	16	200
DialogSum	3e-05	1024	2	200
QAConv	3e-05	2048	4	200
Molweni	3e-05	2048	4	200

Table 8: Main hyperparameters used for training in our experiments. LR denotes learning rate. MT indicates max tokens. GStep represents gradient accumulation step. WStep stands for warmup step.

Dataset	**BeamSize**	**LenP**	**MinLen**	**MaxLen**
QMSum	4	1.0	55	140
SummScreen	10	2.0	256	450
DialogSum	5	0.5	1	100
QAConv	3	0.1	1	20
Molweni	3	0.1	1	20

Table 9: Main settings used for generation in our experiments. LenP denotes length penalty.

A Model Settings

We list the hyperparameters used for training BART and ATM in our experiments in Table 8. The settings of generation are shown in Table 9.

B Evaluation Metrics

For BERTScore (Zhang et al., 2020b)[5], FactCC (Kryscinski et al., 2020)[6], and SUMMAC (Laban et al., 2022)[7], we all use the official implementations to evaluate our models. For BARTScore (Yuan et al., 2021)[8], we employ BART finetuned on CNN/DM (Hermann et al., 2015) to evaluate our models.

C Case Study

As shown in Table 10, we sample several cases of the generated summaries to illustrate the advantages of ATM. The comparison shows that ATM generates more coherent summaries than BART and mitigates unfaithfulness problems, such as coreference error and missing information.

[5] https://github.com/Tiiiger/bert_score

[6] https://github.com/salesforce/factCC

[7] https://github.com/tingofurro/summac

[8] https://github.com/neulab/BARTScore

Dialogue	Person1: Sally,here is a letter for us. It's from Tom. Person2: Can you read it, please? My hands are wet with all this washing. Person1: Well, OK. Dear Sally and John. Thanks for your letter. It was good to hear from you. Just a short note in reply. Please do call me when you arrive so that I can pick you up at the station.
BART	Person1 gives Sally and John a letter from Tom who will be in town in January.
ATM	Tom writes a letter to Sally and John and tells them he will pick them up when they arrive in town in January.
Dialogue	Person1: Good afternoon, what can I do for you? Person2: Yes, please. I would like to know something about the driving courses. Person1: Well, We have short full time courses during the summer. Are you interested in them? Person2: No, I am free only at weekends. Person1: Then there are weekend courses. The course starts at 8:00 ... Person2: Sounds fine. What about the coaches? Person1: We have very excellent coaches here ... Person2: Good. How many hours of training should I have each day? Person1: 3 hours in the morning and 2 in the afternoon. (...)
BART	Person2 wants to know something about the driving courses. Person1 introduces the short full-time courses, weekend courses, and the coaches.
ATM	Person2 wants to know something about the driving courses. Person1 introduces the short full-time courses and the weekend courses and introduces the coaches and the training time.
Dialogue	(...) Person2: This next one is from Betty. Person1: A highchair and car seat! Wow Betty, thank you so much! I really appreciate it! Person2: One more from Carla. Person1:A playpen and crib! Thanks Carla! This is just what I needed! (...) Person1: Umm. I think my water just broke! Get me to a hospital!
BART	Person1 thanks Person2 for organizing a great baby shower. Person1 Person 1 opens some presents from Betty, Carla, Betty, and Carla and feels like having a baby.
ATM	Person1 thanks Person2 for organizing the baby shower and opening some presents, including a bib, a stroller, a highchair, and a car seat from Betty and Carla. Person1's water breaks and asks Person2 to take her to a hospital.

Table 10: ATM and BART outputs for DialogSum dataset. The outputs reflect three main problems based on the corresponding dialogue: (1) coreference error; (2) missing information; (3) redundancy. ATM generates more coherent summaries which avoid these problems effectively and achieves better performance than BART.