TANET: Thread-Aware Pretraining for Abstractive Conversational Summarization

Ze Yang[1], **Liran Wang**[1], **Zhoujin Tian**[1], **Wei Wu**[2], **Zhoujun Li**[1]*
[1]State Key Lab of Software Development Environment, Beihang University, Beijing, China
[2]Meituan, Beijing, China
{tobey,wanglr,eitbar,lizj}@buaa.edu.cn
wuwei19850318@gmail.com

Abstract

Although pre-trained language models (PLMs) have achieved great success and become a milestone in NLP, abstractive conversational summarization remains a challenging but less studied task. The difficulty lies in two aspects. One is the lack of large-scale conversational summary data. Another is that applying the existing pre-trained models to this task is tricky because of the structural dependence within the conversation and its informal expression, etc. In this work, we first build a large-scale (11M) pretraining dataset called RCSUM, based on the multi-person discussions in the Reddit community. We then present TANET, a thread-aware Transformer-based network. Unlike the existing pre-trained models that treat a conversation as a sequence of sentences, we argue that the inherent contextual dependency among the utterances plays an essential role in understanding the entire conversation and thus propose two new techniques to incorporate the structural information into our model. The first is *thread-aware attention* which is computed by taking into account the contextual dependency within utterances. Second, we apply *thread prediction* loss to predict the relations between utterances. We evaluate our model on four datasets of real conversations, covering types of meeting transcripts, customer-service records, and forum threads. Experimental results demonstrate that TANET achieves a new state-of-the-art in terms of both automatic evaluation and human judgment.

1 Introduction

Text summarization is a long-standing challenging task in artificial intelligence, aiming to condense a piece of text to a shorter version, retaining the critical information. There are various promising applications of conversational summarization in the real world, emphasizing the need to build auto summarization systems. For example, online customer-service staff can improve work efficiency by recording the customer demands and current solutions after each communication. In the industry, meeting summaries are also generally required in order to track the progress of projects. The automatic doctor-patient interaction summary can save doctors much time from filling out medical records. Therefore, conversational summarization has been a potential field in summarization and has received increasing attention.

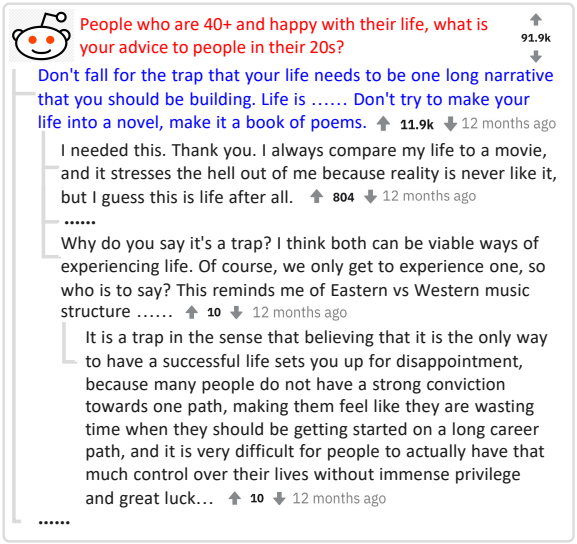

Figure 1: An abbreviated example from RCSUM. It contains a total of 14k comments and more than 210k words in this post. The title and the lead comment are selected as the pseudo summary of this thread.

Benefiting from the availability of large-scale high-quality data, abstractive document summarization has been extensively explored in the past years (Rush et al., 2015; See et al., 2017; Chen and Bansal, 2018). Recently, the pretraining methods further extend the success (Lewis et al., 2020; Zhang et al., 2020). In contrast, abstractive conversational summarization is a more challenging but less studied task. The reason mainly lies in: (1) compared with news, there are no large-scale

* Corresponding Author

Findings of the Association for Computational Linguistics: NAACL 2022, pages 2594 - 2607

publicly available labeled datasets for abstractive conversational summarization; (2) conversations are usually informal, verbose, and repetitive, sprinkled with false starts, backchanneling, reconfirmations, hesitations, and speaker interruptions (Sacks et al., 1978), which makes the whole session difficult to understand; (3) unlike the linear relationship in the one-speaker document, there are always multiple speakers in a conversation, and the inherent contextual relationships are structured; (4) conversations in some scenarios could be much longer than a document. For instance, in CNN/Daily Mail dataset (Hermann et al., 2015), the average number of words in a document is 781, while the average length of the transcripts in ICSI, a widely explored meeting corpus, is 10,189. These challenges encourage us to explore conversation-oriented summarization methods.

To overcome the challenges, we study pretraining for abstractive conversational summarization in this work. To tackle the bottleneck of insufficient data, we first build a large-scale (11M) corpus for conversational summarization called RCSUM, based on the multi-person discussions crawled from the Reddit website. For the absence of the summary of discussions, we propose two heuristic strategies to select the title and the lead comment of a thread as its summary-like sentences. Figure 1 shows an abbreviated example in RCSUM. For the model architecture, we present TANET, a Thread-Aware NETwork for abstractive conversational summarization. As conversations are usually lengthy, we adopt the hierarchical encoders, which consist of a token encoder and an utterance encoder. Unlike the existing pre-trained models that treat a conversation as a sequence of sentences, we argue that the inherent contextual dependency among the utterances plays an essential role in understanding the entire conversation and thus propose two new techniques to incorporate the structural information into TANET. First, we replace the self-attention layers in the utterance encoder with the thread-aware relative attention. Second, we propose a new pretraining task, the thread prediction, to further enhance the representations by predicting the relations across a small set of utterances.

We evaluate TANET on four datasets of conversational summarization, covering domains of meeting transcripts (Carletta et al., 2005; Janin et al., 2003), customer-service records (Yuan and Yu, 2019), and forum threads (Tarnpradab et al., 2017). Experimental results indicate that TANET achieves new state-of-the-art on all datasets in terms of both automatic evaluation and human judgement.

In summary, our contributions in this work are three-fold: (1) We build a large-scale pretraining corpus based on real conversations for abstractive conversational summarization. (2) TANET is the first pre-trained abstractive conversational summarization model with inherent structure modeling. (3) The effectiveness of TANET is demonstrated on four downstream datasets of conversational summarization, covering types of meeting transcripts, customer-service records, and forum threads.

2 Problem Formalization

In general, the abstractive conversational summarization task could be formalized as follows. Denote the dataset as $\mathcal{D} = \{(C_i, S_i)\}_{i=1}^{N}$, where $\forall i$, (C_i, S_i) is a conversation-summary pair and N is the size of $\mathcal{D}$. The conversation $C_i = \{(u_{i,j}, a_{i,j})\}_{j=1}^{n_i}$ consists of n_i rounds of utterances $\{u_{i,j}\}_{j=1}^{n_i}$ and their associated attributes $\{a_{i,j}\}_{j=1}^{n_i}$. For example, each meeting transcript in the AMI dataset comprises multiple turns, where each turn is an utterance of a participant who has a specific role in the project, such as manager or designer. Our goal is to learn a generation probability $P(S|C)$, so that given a new conversation input C, we can generate a summary S .

Since there is always limited availability of $\mathcal{D}$ to support accurately learning for $P(S|C)$, we propose to build a large-scale summarization-like corpus $\mathcal{D}_p = \{(C_k, S_k)\}_{k=1}^{M} (M \gg N)$ by leveraging massive accessible conversation data. S_k represents the pseudo summary of the conversation C_k. In this way, we first pretrain our model on $\mathcal{D}_p$ and then finetune it on the respective dataset $\mathcal{D}$ of each downstream task.

3 Reddit for Conversational Summarization

Existing conversational summarization corpora (Carletta et al., 2005; Janin et al., 2003; Tarnpradab et al., 2017; Yuan and Yu, 2019; Gliwa et al., 2019) have a low number of conversations, which prevents research community from engaging into this problem. Different from (Zhu et al., 2020) that using news documents to simulate multi-person conversations for pretraining, it is reasonable to hypothesize that leveraging real conversation data could lead to better downstream performance. In

this work, to benefit from the large-scale conversation corpus, we mined and processed a large-scale dataset from **R**eddit [1] for **C**onversational **Sum**marization called **RCSUM**. Figure 1 shows an example in the dataset. To our best knowledge, RCSUM is the first large-scale pretraining corpus with real conversations for abstractive conversational summarization.

We collected the posts on the Reddit site from 2019 to 2020. A post is composed of a title and its corresponding discussions which usually consist of multiple threads. The comments in a thread can naturally expand into a tree structure. Remarkably, each comment has rich attributes, including the user information, creation timestamp and the accumulated score[2], etc. With the large-scale real multi-person conversation data, the key is how to construct a summary-like instance for a thread. We consider two strategies to select sentences that appear to dominate the thread: (1) **Title**. The discussions of each post are all developed upon the topic of the title, so we select the title as a part of the pseudo summary; (2) **Lead Comment**. Despite the topic given by the title, the lead comment (i.e., the first comment of a thread) also well influences the future direction of what is discussed in this thread. We concatenate them as the pseudo summary of the discussions in a thread. Lead comment's original position is replaced by a special token [MASK].

To clean up RCSUM, we adopted a series of heuristics including: (1) We removed any threads where the number of comments less than 10; (2) We discarded any not-safe-for-work posts, such as posts containing adult or violent content; (3) We replaced all URLs with a special token [URL]; (4) We removed all markup and any other non-text content such as "*, ∼, [,]"; (5) We removed any threads whose title or lead comment scored less than 0; (6) We removed any posts which contain quarantine, picture, or video, etc. After then, the dataset has $11,200,981$ instances.

4 Methodology

In this section, we present TANET, a thread-aware pretrained model which incorporates the inherent dependencies between utterances to enable improved conversation's representations for summary generation. Below, we first introduce the model architecture, thread-aware attention, and then introduce our pretraining objectives. Finally, we move on to the application of downstream tasks.

4.1 Model Architecture

Encoder. We employ hierarchical encoders, a *token encoder* and an *utterance encoder*, to represent the input conversation. This design mainly comes from two considerations: (1) The conversations in actual applications are lengthy (e.g., The Reddit post in Figure 1 has more than 210k words, and a meeting transcript usually consists of thousands of tokens.), thus it may not be feasible to simply apply the canonical transformer structure. (2) Hierarchical architecture is more suitable for the conversational tasks to carry out modeling of utterances and interactive structure of the conversation.

Let $C = (u_0, \cdots, u_{|C|})$ denote an conversation instance in the pretraining corpus $\mathcal{D}_p$. $u_i = (\langle \texttt{bos} \rangle, w_{i,1}, \cdots, w_{i,|u_i|})$ is the token sequence of $i\text{-}th$ utterance after tokenization, where $\langle \texttt{bos} \rangle$ is a special token in vocabulary $\mathcal{V}$ to represent the beginning of a turn. The token encoder takes each sequence u_i as the input and first converts it into input vectors $\mathbf{H}_i^{\mathcal{T},0} \in \mathbb{R}^{|u_i| \times d_h}$. For each token, its input vector is constructed by summing up the corresponding token embedding and the sine-cosine positional embedding (Vaswani et al., 2017). Then, N identical layers are nested over $\mathbf{H}_i^{\mathcal{T},0}$ to produce the contextual representations by:

$$\mathbf{H}_i^{\mathcal{T},N} = \text{Transformer}^{\mathcal{T}}(\mathbf{H}_i^{\mathcal{T},0}) \quad (1)$$

Each layer consists of two sub-layers, a self-attention sub-layer followed by a position-wise feed-forward sub-layer and uses residual connections around each of them. We adopt the pre-layer normalization following several recent works (Baevski and Auli, 2019; Child et al., 2019; Wang et al., 2019; Xiong et al., 2020), which place the layer normalization inside the residual connection. That is, given input x, the output of each sub-layer is $x + \text{Sublayer}(\text{LayerNorm}(x))$. The utterance encoder also has N identical transformer layers in structure, which processes the information at turn level. All utterances are arranged in the order of their timestamps, and we employ the sine-cosine positional embedding to model the chronological order. Let $\mathbf{H}^{\mathcal{U},0} = (h_{u_0}, \cdots, h_{u_{|C|}})$ denotes the sequence of representations of utterances. For the i-th turn u_i, the embedding of $\langle \texttt{bos} \rangle$ is chosen as its representation, i.e., $h_{u_i} = \mathbf{H}_{i,0}^{\mathcal{T},N}$. Different from

[1] https://www.reddit.com

[2] The score is the number of upvotes minus the number of downvotes.

the token encoder, we propose the Thread-Aware Attention sub-layer to replace the self-attention sub-layer to encode the tree-structure information into our model.

Thread-Aware Attention. Each sub-layer consists of h attention heads, and the results from each head are concatenated together and projected to form the output of the sub-layer. Formally, given the input $\mathbf{H}^{\mathcal{U},0}$, the k-th head computes a new sequence $z_k = (z_{k,0}, \cdots, z_{k,|C|})$ by:

$$z_{k,i} = \sum_{j=1}^{|x|} \alpha_{ij}(h_{u_j} W_k^V), \qquad \alpha_{ij} = \frac{\exp e_{ij}}{\sum_{t=1}^{|x|} \exp e_{it}} \tag{2}$$

where $z_{k,i} \in \mathbb{R}^{d_z}, d_z = d_h/h$. e_{ij} is the attention weight from h_{u_j} to h_{u_i}. Inspired by the relative position encoding (RPE) works (Shaw et al., 2018; Huang et al., 2020), we consider the interactions of queries, keys, and relative positions simultaneously to fully utilize the structural information of a conversation:

$$e_{ij} = \frac{(h_{u_i} W_k^Q + r_{ij})(h_{u_j} W_k^K + r_{ij})^\top - r_{ij} r_{ij}^\top}{\sqrt{d_z}} \tag{3}$$

where $W_k^Q, W_k^K, W_k^V \in \mathbb{R}^{d_h \times d_z}$ are parameter matrices. $\sqrt{d_z}$ is a scaling factor for stable training. The key to this mechanism is that $r_{i,j} \in \mathbb{R}^{d_z}$ encodes the relation from utterance u_j to u_i, which is defined as:

$$r_{i,j} = \begin{cases} w_{\text{clip}(\text{depth}(u_i) - \text{depth}(u_j), k)}, & 1) \\ w_*, & 2) \end{cases} \tag{4}$$

As illustrated in Figure 2, the relation between two utterances has two situations: 1) one is a parent or child utterance of the other, that is, they belong to the same path, e.g. u_1 and u_4; 2) otherwise, e.g. u_2 and u_3. We totally define $2k + 2$ learnable thread-aware position embeddings $\{w_*, w_{-k}, \cdots, w_k\}$. where $\text{clip}(x, k) = \max(-k, \min(k, x))$, and the function $\text{depth}(u_i)$ returns the distance between utterance u_i and the first utterance u_0 in the thread, e.g. $\text{depth}(u_4) = 2$. The output of the utterance encoder is $\mathbf{H}^{\mathcal{U},N} \in \mathbb{R}^{|C| \times d_h}$.

Decoder. The decoder is a N-layer transformer to generate the summary S. At the training stage, the decoder takes the right-shifted token sequence of S as input. In each layer, the self-attention sub-layer leverages a lower triangular mask to prevent positions from attending to their future positions. Then, the cross-attention sub-layers attend with the outputs from the hierarchical encoder. In particular, we make an encoder-wise residual connection around the utterance encoder to propagate the token-level information directly to the decoder. We found that this can improve the model's capability to reproduce the words involved in the conversation. Denote the output of the decoder as $\mathbf{H}^{\mathcal{D},N} \in \mathbb{R}^{|S| \times d_h}$. When predicting the i-th token s_i, we reuse the embedding matrix of the vocabulary $\mathcal{E}_\mathcal{V} \in \mathbb{R}^{|\mathcal{V}| \times d_h}$ to project $\mathbf{H}_{i-1}^{\mathcal{D},N}$ into a probability distribution:

$$P(S_i | S_{<i}, C) = \text{Softmax}(\mathbf{H}_{i-1}^{\mathcal{D},N} \mathcal{E}_\mathcal{V}^\top) \tag{5}$$

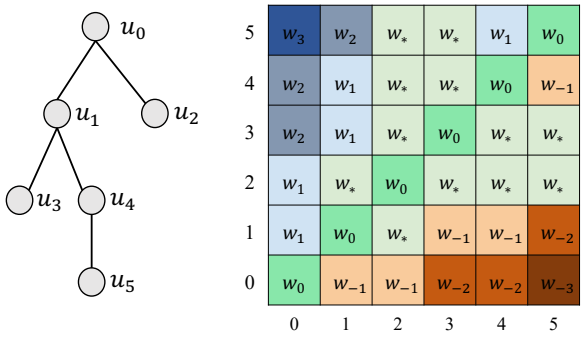

Figure 2: Illustration of Thread-Aware attention. The left is the tree structure of a conversation thread, and u_i represents the i-th utterance. The thread-aware attention weights across the utterances are on the right.

4.2 Pretraining Objectives

In this section, we describe the pretraining objectives used for pretraining TANET. In addition to the causal language modeling, we newly introduce another thread-aware pretraining task to predict the contextual relation between utterances.

Causal Language Modeling. Following many previous works (Lewis et al., 2020; Zhang et al., 2020), we apply the causal language modeling objective, which seeks to minimize the cross-entropy loss:

$$\mathcal{L}_{CLM}(\theta) = -\frac{1}{|S|} \sum_{i=1}^{|S|} \log P(S_i | S_{<i}, C) \tag{6}$$

Thread Prediction. To enhance the representation of the thread structure in a conversation, we introduce a new pretraining task of thread prediction. The motivation is to encourage the model

to learn thread-aware representations that encode the information of which comments this one was written based on, that is, its historical comments. Specifically, we randomly sample 20% utterances C_s from C and then let the model predict their historical comments. Formally, the pretraining objective is calculated as:

$$\mathcal{L}_{ThreadPred} = -\sum_{a_{ij}\in\mathcal{A}} \Big(\delta(a_{ij}) \log p_{a_{ij}} + \big(1-\delta(a_{ij})\big)\log(1-p_{a_{ij}})\Big) \quad (7)$$

where $\mathcal{A} = C_s \times C \cup C \times C_s$ is the set of comment pair candidates for prediction. $\delta(a_{ij})$ returns 1 if u_j and u_i belong to one thread and u_j is history of u_i, otherwise 0. p_{ij} is the probability of u_j being the historical comment of u_i and is computed by:

$$p_{a_{ij}} = \text{Sigmoid}\big((\mathcal{H}_{i,0}^{\mathcal{T},N} W_a)(\mathcal{H}_{j,0}^{\mathcal{T},N} W_b)^\top\big) \quad (8)$$

$W_a, W_b \in \mathbb{R}^{d_h \times d_h}$ are two parameter matrices.

4.3 Application on Downstream Tasks

After pretraining on RCSUM, we finetune our model on the downstream tasks. Different tasks will have some differences in data annotation that requires us to adapt it flexibly. For example, the interdependencies among the utterances in a meeting are not labeled in AMI and ICSI, so we treat them as a sequence arranged by time. Besides, some additional information is essential for the generation of summary, which can be prompted to the model by modifying the input utterances. For example, the name and role of each participant are useful for meeting summarization in AMI and ICSI. Without changing the model structure, we inform TANET of the information by replacing the original utterance with template "{*participant*} of role {*role*} said: {*utterance*}".

5 Experiments

5.1 Datasets

We evaluate TANET and all baseline models on four benchmark datasets of long and real-life conversations, covering domains of meeting transcripts, customer-service records, and threads in web forum. Table 1 summarizes the statistics of the four datasets.

AMI (Carletta et al., 2005) is a multi-modal dataset consisting of 100 hours of meeting recordings with rich annotations. Following Shang et al. (2018); Zhu et al. (2020), we select 137 meetings of scenario where the participants play different roles in a design team. Each meeting is labeled with transcripts produced by automatic speech recognition (ASR) and an abstractive summary written by a human annotator. Furthermore, each dialogue is also associated with additional information, including its speaker id with role, dialogue act. We use the same data split of 100/17/20 as training/validation/test sets.

Dataset	AMI	ICSI	MultiWOZ	FORUM
Domain	*Meeting*	*Meeting*	*Customer Service*	*Forum Thread*
# Speakers	4	6.2	2	6.8
# Conversations	137	59	10,438	689
# Conv. words	4,757	10,189	180.7	825.0
# Summ. words	322	534	91.9	190.6
# Turns	289	464	13.7	10.5

Table 1: Statistics of the converstional summarization datasets. The number of conversation words, summary words, turns and speakers are all averaged across all conversations in the dataset.

ICSI (Janin et al., 2003) is another widely-used meeting corpus consisting of about 70 hours of meeting audio recordings with orthographic transcription and other manual annotations. We follow the pre-processing pipeline from Zhu et al. (2020) and split the training/validation/test sets of size 43/10/6, respectively. Each meeting also contains a manually labeled abstractive summary and the associated role information for each participant.

MultiWOZ (Yuan and Yu, 2019) is an abstractive dialog summarization dataset based on the MultiWOZ corpus (Budzianowski et al., 2018; Ramadan et al., 2018; Eric et al., 2019; Zang et al., 2020), which is a fully-labeled collection of human-human written conversations spanning over multiple domains and topics. The dataset is built on various customer-service records in the corpus, such as booking restaurants, hotels, taxis. We use the summary annotation provided by Yuan and Yu (2019), and the same data split of 8438/1000/1000 as training/validation/test sets.

FORUM (Tarnpradab et al., 2017) contains 700 human-annotated forum threads. Each thread contains a human-annotated abstractive summary and multiple posts written by several different users. These threads are collected from tripadvisor.com and ubuntuforums.org. Bhatia et al. (2014) annotated 100 threads from TripAdvisor with human-writtern summaries, and Tarnpradab et al. (2017) further extend the summary annotation with 600 more threads. In our experiments, we divide

the dataset into 500/100/89 examples for training/validation/test sets.

5.2 Metrics

ROUGE (Lin, 2004) is a standard metric for summarization task. Following Zhu et al. (2020), we use ROUGE-1, ROUGE-2, and ROUGE-SU4 to evaluate all meeting summarization models. The models on MultiWOZ and FORUM are evaluated by ROUGE-1, ROUGE-2 and ROUGE-L (Tarnpradab et al., 2017; Yuan and Yu, 2019). We obtain the scores by the rouge-metric package [3].

5.3 Evaluation Results and Discussions

5.3.1 Meeting Summarization

We compare TANET with a variety of models from previous literature: **Random** (Riedhammer et al., 2008), the template-based model **Template** (Oya et al., 2014), two ranking systems **TextRank** (Mihalcea and Tarau, 2004) and **ClusterRank** (Garg et al., 2009), the unsupervised method **UNS** (Shang et al., 2018), **Extractive Oracle**, which concatenates top sentences with the highest ROUGE-1 scores with the golden summary, the document summarization model **PGNet** (See et al., 2017), **Copy from Train**, which randomly copies a summary from the training set as the prediction, the multimodal model **MM** (Li et al., 2019), and the hierarchical Network **HMNet** (Zhu et al., 2020). Besides the baselines above, **BART** (Lewis et al., 2020) and **PEGASUS** (Zhang et al., 2020), two state-of-the-art pre-trained models on document summarization, and **Longformer-Encoder-Decoder** (**LED**) (an, 2020) are also included in comparison to have a thorough understanding towards our model. We concatenate all turns of a transcript into a sequence and then truncate it to meet the length constraints of the model input. LED_{large} is initialized from BART_{large} and able to process 16k tokens. Please refer to the Appendix for more implementation details .

Table 2 reports the automatic evaluation results on datasts AMI and ICSI. We can see that, except for ROUGE-1 on AMI, TANET outperforms all baseline models in all metrics. MM is a multimodal model which requires additional annotation of topic segmentation (TopicSeg) and multi-modal features derived from the visual focus of attention (VFOA) collected by cameras. In practice, the visual information is rarely available, such as online chat, so the application scenarios of MM are very limited. In comparison, TANET is completely based on meeting transcripts from ASR systems, so it has better scalability. Comparable performance is achieved in ROUGE-1 on AMI, but it is significantly higher in ROUGE-2 by 7.2 points. In particular, TANET outperforms HMNet, indicating that pretraining on large-scale conversation data while incorporating the inherent structural information can lead to better performances on downstream tasks. Moreover, TANET significantly outperforms BART and PEGASUS on both AMI and ICSI. Although the two baselines own strong capabilities to summary a document, the tricky part is that a meeting transcript is very long and cannot be fully fed into the models. For example, the average number of words in ICSI is 10,189, which far exceeds

Models	R-1	R-2	R-SU4
AMI			
Random (Riedhammer et al., 2008)	35.13	6.26	13.17
Template (Oya et al., 2014)	31.50	6.80	11.40
TextRank (Mihalcea and Tarau, 2004)	35.25	6.90	13.62
ClusterRank (Garg et al., 2009)	35.14	6.46	13.35
UNS (Shang et al., 2018)	37.86	7.84	14.71
Extractive Oracle	39.49	9.65	13.20
PGNet (See et al., 2017)	40.77	14.87	18.68
Copy from Train	43.24	12.15	14.01
MM+TopicSeg (Li et al., 2019)†	51.53	12.23	-
MM+TopicSeg+VFOA (Li et al., 2019)†	**53.29**	13.51	-
HMNet (Zhu et al., 2020)	53.02	18.57	24.85
Our re-implementation			
LED_{large} (an, 2020)	53.10	19.83	24.95
BART_{base} (Lewis et al., 2020)	50.26	18.18	17.83
PEGASUS_{large} (Zhang et al., 2020)	47.05	16.64	16.03
TANET (*ours*)	53.26	**20.73***	**25.98***
ICSI			
Random (Riedhammer et al., 2008)	29.28	3.78	10.29
TextRank (Mihalcea and Tarau, 2004)	29.70	4.09	10.64
ClusterRank (Garg et al., 2009)	27.64	3.68	9.77
UNS (Shang et al., 2018)	31.60	4.83	11.35
Extractive Oracle	34.66	8.00	10.49
PGNet (See et al., 2017)	32.00	7.70	12.46
Copy from Train	34.65	5.55	10.65
HMNet (Zhu et al., 2020)	46.28	10.60	19.12
Our re-implementation			
LED_{large} (an, 2020)	43.13	11.76	19.08
BART_{base} (Lewis et al., 2020)	42.01	9.96	11.72
PEGASUS_{large} (Zhang et al., 2020)	42.44	9.15	11.10
TANET (*ours*)	**47.21***	**12.35***	**19.27**

Table 2: Automatic evaluation results on datasets AMI and ICSI. Numbers in bold indicate the best performing models on the corresponding metrics. Numbers marked with "*" mean that the improvement over the best baseline is statistically significant (t-test with p-value < 0.05). Models marked with "†" require additional human annotations of topic segmentation and visual signals from cameras.

[3] https://pypi.org/project/rouge-metric/

Models	R-1	R-2	R-L
PGNet (See et al., 2017)	62.89	48.61	59.30
Transformer (Vaswani et al., 2017)	63.12	50.63	61.04
SPNet (Yuan and Yu, 2019)	90.97	84.14	85.00
Our re-implementation			
LED_{large} (an, 2020)	91.41	79.93	83.63
HMNet (Zhu et al., 2020)	66.33	50.49	64.52
BART_{base} (Lewis et al., 2020)	81.47	70.24	73.14
PEGASUS_{large} (Zhang et al., 2020)	**93.51**	88.09	84.73
TANET (*ours*)	93.25	**88.60***	**85.67***

Table 3: Automatic evaluation results on MultiWOZ. Numbers in bold indicate the best performing models on the corresponding metrics. Numbers marked with "*" mean that the improvement over the best baseline is statistically significant (t-test with p-value < 0.05).

Models	R-1	R-2	R-L
ILP (Berg-Kirkpatrick et al., 2011)	29.3	9.9	-
Sum-Basic (Vanderwende et al., 2007)	33.1	10.4	-
KL-Sum	35.5	12.3	-
Lex-Rank (Erkan and Radev, 2011)	38.7	14.2	-
MEAD (Radev et al., 2004)	38.5	15.4	-
SVM (Chang and Lin, 2011)	24.7	10.0	-
LogReg (Fan et al., 2008)	29.4	7.8	-
HAN (Tarnpradab et al., 2017)	37.8	14.7	-
Our re-implementation			
LED_{large} (an, 2020)	42.39	22.78	30.48
HMNet (Zhu et al., 2020)	41.30	17.12	31.76
BART_{base} (Lewis et al., 2020)	42.91	22.32	30.35
PEGASUS_{large} (Zhang et al., 2020)	42.92	20.50	29.16
TANET (*ours*)	**45.20***	**25.61***	**33.59***

Table 4: Automatic evaluation results on FORUM. Numbers in bold indicate the best performing models on the corresponding metrics. Numbers marked with "*" mean that the improvement over the best baseline is statistically significant (t-test with p-value < 0.05).

the maximum input length 512 tokens of BART and PEGASUS. As a result, most of the content in a meeting transcript are discarded, which will inevitably limit the performances of the two models. LED can input all sentences, but it is still difficult to fully understand the conversation, which further demonstrate the effectiveness of the pretraining on the corpus RCSUM.

5.3.2 Customer-service Records Summarization

To demonstrate the effectiveness of TANET on customer-service records summarization, following models are selected as baselines from previous literature: the pointer-generator network **PGNet** (See et al., 2017), **Transformer** (Vaswani et al., 2017) and **SPNet** which incorporates three types of semantic scaffolds - speaker role, semantic slot and dialogue domain for summarization (Yuan and Yu, 2019). Besides, we include **HMNet** (Zhu et al., 2020) as a baseline and implement it using the official code [4]. We also apply **Longformer-Encoder-Decoder (LED)** (an, 2020), **BART** (Lewis et al., 2020) and **PEGASUS** (Zhang et al., 2020) in this task by concatenating all utterances in a conversation as a document.

Table 3 reports the automatic evaluation results on MultiWOZ. We can observe that TANET achieves new state-of-the-art performance on ROUGE-2 and ROUGE-L, which demonstrate the effectiveness of pretraining on large-scale conversation data. Different from the results of the meeting summarization given in Table 2, PEGASUS_{large} achieves close performance to TANET and even the best in ROUGE-1, showing its great generalization ability in this task. This is because: (1) the "documents" can be fully fed into the model without content loss. The average length of the dialogue in MultiWOZ is 180.7 words, which do not exceed the model's maximum input length 512. (2) each conversation takes place between two speakers (i.e. a customer and a staff), so the structure of a dialogue can be viewed as a sequence, which is similar to the sentences in a document. Compared with LED, HMNet and BART, the un-pretrained model SPNet obtains surprisingly better scores. This motivates us to combine richer conversation-related information, such as speaker role, dialogue act, semantic slot, and dialogue domain, to further improve model's summarization capabilities in the future.

5.3.3 Forum Threads Summarization

In this task, TANET is compared against a range of baselines, including following unsupervised methods: (1) **ILP** (Berg-Kirkpatrick et al., 2011), a baseline integer linear programming framework; (2) **Sum-Basic** (Vanderwende et al., 2007), a model that assumes words occurring frequently in a document cluster have a higher chance of being included in the summary; (3) **KL-Sum**, an approach that select the sentences decreasing the KL divergence as the summary; (4) **Lex-Rank** (Erkan and Radev, 2011), a graph-based model based on eigenvector centrality; (5) **MEAD** (Radev et al., 2004), a centroid-based approach which scores sentences based on length, centroid, and position; and supervised extractive systems, including (1) **SVM**

[4] https://github.com/microsoft/HMNet

Models	AMI			FORUM		
	R-1	R-2	R-SU4	R-1	R-2	R-L
TANET	53.26	20.73	25.98	45.20	25.61	33.59
- Pretraining	46.43	16.85	18.42	39.67	15.37	26.45
- Thread-Aware Attention	51.33	18.90	23.71	41.85	22.41	30.79
- Thread Prediction	51.94	19.30	24.75	41.70	21.33	31.80
- Encoder-wise Residual	51.86	20.02	24.59	44.93	24.78	33.30

Table 5: Ablation study on AMI and FORUM.

Models	AMI			FORUM		
	Read.	Conc.	Kappa	Read.	Conc.	Kappa
HMNet	1.67	1.40	0.60	1.76	1.57	0.61
Bart_{base}	1.58	1.02	0.72	1.82	1.63	0.69
TANET	1.70	1.53	0.63	1.84	1.70	0.60

Table 6: Human evaluation results on AMI and FORUM. "Read.", "Conc." are abbreviations for readability and conciseness, respectively.

(Chang and Lin, 2011), the support vector machine; (2) **LogReg** (Fan et al., 2008), the logistic regression; (3) **HAN** (Tarnpradab et al., 2017), a hierarchical attention network with redundancy removal process; Besides, we also apply the cross-domain pre-trained model **HMNet** (Zhu et al., 2020) in this task and implement **Longformer-Encoder-Decoder** (**LED**) (an, 2020), **BART** (Lewis et al., 2020) and **PEGASUS** (Zhang et al., 2020) in a similar way to the adaptation in the above two tasks.

Table 4 reports the automatic evaluation results on the dataset FORUM. TANET outperforms all baseline models in terms of all metrics, and the improvements are statistically significant (t-test with p-value< 0.05), which further demonstrate the effectiveness of our method. In this task, the gains over the pre-trained baselines are relatively high , due to (1) the consistency of conversation domain between the pretraining stage and downstream fine-tuning. The conversation in FORUM and our pre-training corpus RCSUM are both forum threads. Note that, although the data in FORUM is collected from TripAdvisor (tripadvisor.com) and Ubuntu-Forums (ubuntuforums.org.), the subjects are also included in some specific sub-reddits on the Reddit website. In contrast, LED, HMNet, BART and PEGASUS are all pre-trained with document-like text, so there will be a domain gap in thread understanding; (2) structure modeling. The tree-like reply relationship in a thread plays a vital role in understanding the entire thread, but the baselines can only process it linearly, which poses challenges for the model to fully understand the context and generate accurate summaries.

5.3.4 Ablation Study

To understand the impact of our pretraining strategies on model performance, we compare the full TANET with the following variants: (1) - *Pretraining*: the pretraining stage is removed; (2) - *Thread-Aware Attention*: the Thread-Aware Attention sublayers in the utterance encoder degenerate into standard self-attention sublayers; (3) -*Thread Prediction*: $\mathcal{L}_{ThreadPred}$ is removed; and (4) - *Encoder-wise Residual*: the encoder-wise residual around the utterance encoder is removed. Table 5 reports the evaluation results on AMI and FORUM.[5] We can conclude that (1) the pretraining on RCSUM helps to significantly improve the performance, as removing it results in dramatic performance drop on both AMI and FORUM; (2) both Thread-Aware attention and Thread Prediction objective are useful, indicating that the structure of thread is essential to facilitate the understanding of conversation, especially for the threads with tree structure; (3) the encoder-wise residual is meaningful, as removing it causes performance drop.

5.3.5 Human Evaluation

As the human annotation for this task is very time-consuming and labor-intensive, we also conduct human evaluation on the test sets of AMI and FORUM to verify whether the improvements on automatic evaluation is in line with the human perceived quality. We recruit 3 well-educated native speakers as annotators and compare TANET with BART_{base} and HMNet on 2 aspects - *readability* and *conciseness*. The former measures how fluent a generated summary is, while the later measures how well the summary sums up the main ideas of a conversation. For each sample, we show its conversation, reference summary, as well as summaries generated by models (the order is shuffled to hide their sources) to the annotators and ask them to judge the quality and assign a score in {0,1,2} (indicating "bad", "fair", and "good") to each summary for each aspect. Table 6 reports the evaluation results. We can observe (1) the three models are comparable on readability on both datasets; (2) TANET outperforms the others on conciseness, which is consistent with the automatic evaluation results in Table 2 and Table 4; (3) Bart_{base} does not perform well on conciseness on AMI, as most utterances of a conversation are discarded due to the input length

[5] Ablation results on AMI and FORUM could provide more insights, whereas ICSI is similar to AMI, and MultiWOZ is less challenging.

constraint 512. All kappa values are no less than 0.6, indicating substantial agreement among the annotators. For reference, we present case study in the Appendix.

6 Related Work

With the recent success of seq2seq models, the research focus of conversational summarization has been transferred from the extractive methods to abstractive models. Various semantic patterns have been applied to these abstractive approaches, such as dialogue acts (Goo and Chen, 2018), auxiliary key point sequences (Liu et al., 2019a), topic segments (Li et al., 2019; Liu et al., 2019b), conversational stages and dialogue overview (Chen and Yang, 2020), discourse relations (Murray et al., 2006; Bui et al., 2009; Qin et al., 2017). At the same time, some work is devoted to providing high-quality datasets to promote the development of this research direction (Carletta et al., 2005; Janin et al., 2003; Tarnpradab et al., 2017; Yuan and Yu, 2019; Gliwa et al., 2019). However, these corpora have a low number of conversations, which hinders the progress of abstractive summarization (an, 2021). Recently, large neural models pre-trained on huge corpora have led to strong improvements on numerous natural language understanding and generation tasks (Devlin et al., 2019; Radford et al., 2019; Lewis et al., 2020; Zhang et al., 2020). Encouraging by the promising progress of pretraining, Zhu et al. (2020) first introduce a hierarchical structure and propose pretraining on cross-domain data for meeting summarization. The pretraining data is collected from the news domain. Regarding one document as the utterances from one participant, multiple documents are combined and reshuffled to simulate a multi-person meeting. However, there are two disadvantages - the first is the style inconsistency between conversation and news, and the second is that there is no contextual relationship between the two documents, so the participants have no communication actually. In this work, we build a large-scale corpus based on real conversations. Besides, we further incorporate the structure information of thread in the model.

7 Conclusions

In this work, we introduce TANET, a thread-aware pre-trained model for abstractive conversational summarization. TANET employ the thread-aware attention and a new pretraining objective to fully leverage the structure information of conversation. Furthermore, we build a large-scale pretraining corpus based on the discussions on Reddit. Experiments on four downstream tasks demonstrate the effectiveness of TANET.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos.U1636211, 61672081,61370126), the 2020 Tencent Wechat Rhino-Bird Focused Research Program, and the Fund of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2021ZX-18).

References

Iz Beltagy an. 2020. Longformer: The long-document transformer. *ArXiv preprint*, abs/2004.05150.

Xiachong Feng an. 2021. A survey on dialogue summarization: Recent advances and new frontiers. *ArXiv preprint*, abs/2107.03175.

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 481–490, Portland, Oregon, USA. Association for Computational Linguistics.

Sumit Bhatia, Prakhar Biyani, and Prasenjit Mitra. 2014. Summarizing online forum discussions – can dialog acts of individual messages help? In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2127–2131, Doha, Qatar. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Trung Bui, Matthew Frampton, John Dowding, and Stanley Peters. 2009. Extracting decisions from multi-party dialogue using directed graphical models and semantic similarity. In *Proceedings of the*

SIGDIAL 2009 Conference, pages 235–243, London, UK. Association for Computational Linguistics.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. 2005. The ami meeting corpus: A pre-announcement. In *Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction*, MLMI'05, Berlin, Heidelberg. Springer-Verlag.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27.

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.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *ArXiv preprint*, abs/1904.10509.

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.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *ArXiv preprint*, abs/1907.01669.

Günes Erkan and Dragomir R. Radev. 2011. Lexrank: Graph-based lexical centrality as salience in text summarization. *CoRR*, abs/1109.2128.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874.

Nikhil Garg, Benoît Favre, Korbinian Riedhammer, and Dilek Hakkani-Tür. 2009. Clusterrank: a graph based method for meeting summarization. In *INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association, Brighton, United Kingdom, September 6-10, 2009*, pages 1499–1502. ISCA.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. 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, Tomás Kociský, 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 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. 2020. Improve transformer models with better relative position embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3327–3335, Online. Association for Computational Linguistics.

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.

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.

Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196, Florence, Italy. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summariza-*

tion Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019a. Automatic dialogue summary generation for customer service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1957–1965. ACM.

Zhengyuan Liu, Angela Ng, Sheldon Lee, Ai Ti Aw, and Nancy F. Chen. 2019b. Topic-aware pointer-generator networks for summarizing spoken conversations. In *ASRU2019*, pages 814–821.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Gabriel Murray, Steve Renals, Jean Carletta, and Johanna Moore. 2006. Incorporating speaker and discourse features into speech summarization. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 367–374, New York City, USA. 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.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Kechen Qin, Lu Wang, and Joseph Kim. 2017. Joint modeling of content and discourse relations in dialogues. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 974–984, Vancouver, Canada. Association for Computational Linguistics.

Dragomir R. Radev, Hongyan Jing, Magorzata Sty, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Inf. Process. Manag.*, 40(6):919–938.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437, Melbourne, Australia. Association for Computational Linguistics.

Korbinian Riedhammer, Daniel Gillick, Benoît Favre, and Dilek Hakkani-Tür. 2008. Packing the meeting summarization knapsack. In *INTERSPEECH 2008, 9th Annual Conference of the International Speech Communication Association, Brisbane, Australia, September 22-26, 2008*, pages 2434–2437. ISCA.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1978. A simplest systematics for the organization of turn taking for conversation. In *Studies in the organization of conversational interaction*, pages 7–55. Elsevier.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

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.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Sansiri Tarnpradab, Fei Liu, and Kien A. Hua. 2017. Toward extractive summarization of online forum discussions via hierarchical attention networks. In *FLAIRS 2017, Marco Island, Florida, USA, May 22-24, 2017*, pages 288–292. AAAI Press.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Inf. Process. Manag.*, 43(6):1606–1618.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR.

Lin Yuan and Zhou Yu. 2019. Abstractive dialog summarization with semantic scaffolds. *ArXiv preprint*, abs/1910.00825.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

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. Association for Computational Linguistics.

A Implementation Details

In TANET, the token encoder, utterance encoder, and decoder all have 6 layers, i.e., $N = 6$. Each multi-head attention sub-layer has 12 heads, i.e., $h = 12$. The size of feed-forward layer is 3072. The hidden size d_h is 768. We employ the same vocabulary as BART (Lewis et al., 2020), which has 50265 tokens. TANET has 180M parameters in total. We use a dropout probability of 0.1 for all layers. In the thread-aware attention layer, we define 20 learnable embeddings, i.e., $k = 9$. For optimization, both pretraining and downstream finetuning use AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-8$. We pre-train TANET with an accumulated batch size of 256. The initial learning rate is set as $5e-5$ and linearly decreased to 0 after 500k steps. We use beam search with the commonly used trigram blocking (Paulus et al., 2018; Lewis et al., 2020) to select the best candidate during inference for the downstream tasks. To improve the pretraining efficiency, we set the maximum number of utterances to 124, each utterance has a maximum of 200 tokens, and the pseudo-summary has a maximum of 256 tokens. BART_{base} and PEGASUS_{large} are implemented with the codes provided by HuggingFace at `https://github.com/huggingface/transformers/tree/v4.1.1/examples/seq2seq`. The initialized pre-trained models are available at `https://huggingface.co/facebook/bart-base` and `https://huggingface.co/google/pegasus-large`. We implement LED_{large} by referring to the code at `https://colab.research.google.com/github/patrickvonplaten/notebooks/blob/master/Fine_tune_Longformer_Encoder_Decoder_(LED)_for_Summarization_on_pubmed.ipynb`. All experiments are conducted on a machine with 4 NVIDIA Tesla V100 GPUs.

B Case Study

Finally, we randomly sample two cases from the test sets of MultiWOZ and FORUM to conduct qualitative analysis. We show the conversation and the summaries generated by HMNet, LED_{large}, Bart_{base}, TANET, as well as the ground truth in Table 7 and Table 8. We can see that TANET can produce a more concise summary with less incorrect facts and has a higher level of abstraction ability about the conversations. In contrast, the summaries generated by HMNet and Bart_{base} cannot cover all the important information and are prone to some mistakes. LED_{large} tends to copy content from the original conversation, especially as in Table 8.

	Conversation
USER:	I'm looking for information on Cambridge University Botanic Gardens.
SYSTEM:	They are on bateman st, postal code cb21jf, they can be reach at 01223336265, the entrance fee is 4 pounds, can I help with anything else?
USER:	Yes can you help me find a restaurant?
SYSTEM:	The Botanic Gardens are in the centre. Would you like the restaurant to also be in the center? Do you have any type of cuisine in mind?
USER:	Never mind, I will worry about food later. I am actually looking for a hotel with a guesthouse and free parking would be great as well.
SYSTEM:	There are 21 guesthouses with free parking, do you have a price or area preference?
USER:	Cheap and in the south please.
SYSTEM:	I found a guest house listing for Rosa's Bed and Breakfast in the south. Would you like more information?
USER:	I really need a hotel in the west please.
SYSTEM:	i have 1 hotel in the west. the cambridge belfry is cheap with 4 stars. it offers free parking and wifi. do you need a room?
USER:	I'm sorry.I really was looking for a guesthouse.
SYSTEM:	Okay. And did you want it in the west or the south?
USER:	I would like the west please.
SYSTEM:	I found finches bed and breakfast matching your request. Would you like to book a room now?
USER:	Can you please book for Saturday, 2 people, 5 nights please?
SYSTEM:	Booking was successful. Reference number is : W1QIVTNY .
USER:	Thank you so much. I believe that is all I need today. Thanks again for all your help.
SYSTEM:	Have a nice day.

	Summaries
HMNet:	you are traveling to and excited about seeing local tourist attractions . you are looking for a place to stay . the hotel should be in the type of guest house and should include free parking . the restaurant should serve [att_name] food .make sure you get price range , area , and star of the hotel .
LED$_{large}$:	you are traveling to cambridge and looking forward to try local restaurants. you are looking for a particular attraction. its name is called [att_name]. make sure you get address. you you are also looking for information in cambridge. you're looking for places to go in town. the attraction should be in the type of museum. make sure to get postcode. you also are looking to book a hotel. the hotel should be a guest house.
Bart$_{base}$:	you are traveling to cambridge and excited about seeing local tourist attractions. you are looking for a particular attraction. its name is called [att_name] . make sure you get address. you are also looking for a place to stay. the hotel should include free parking and should be in the type of guest house. the hotel should be in the south and should be in the cheap price range. once you find the hotel you want to book it for [people_num] and [hotel_stay] starting from [week_day] . make sure you get the reference number.
TANET:	you are looking for information in cambridge. you are looking for a particular attraction. its name is called [att_name] . make sure you get area and phone number. you are also looking for a place to stay. the hotel should include free parking and should be in the type of guest house. the hotel should be in the cheap price range. if there is no such hotel, how about one that is in the west. once you find the hotel you want to book it for [people_num] and [hotel_stay] starting from [week_day] . make sure you get the reference number.
Ground Truth:	you are looking for information in cambridge. you are looking for a particular attraction . its name is called [att_name] . make sure you get phone number and entrance fee. you are also looking for a place to stay . the hotel should be in the type of guest house and should include free parking. the hotel should be in the west. once you find the hotel you want to book it for [people_num] and [hotel_stay] starting from [week_day] . if the booking fails how about [hotel_stay] . make sure you get the reference number.

Table 7: A case from MultiWOZ. We underline some vital facts in the conversation. Red denotes the incorrect content in the generated summaries. Blue indicates what appears in TANET's summary but is not covered by the ground truth.

	Conversation
N16E:	Hi, I'm hoping a local expert can help us out, we're traveling over to New York (on route to Florida) on Wednesday 28th March, flying out on Saturday 31st, this gives us around 2 and a half days to see the city, below is the list of places we are looking to visit/see. My only thoughts at the moment are to go to the ESB first thing around 8am and TOTR around dusk. I was hoping to use the subway to get around and our hotel is The Belvedere just off 8th Ave on 48th street, in which order should we visit these sights? is it possible? any info on which subway lines to take would be fantastic. We are two families of 4 - 4 adults 4 kids, aged 7 to 13. Maceys - browse for say 2 hours 5th Ave - stroll down and people watch Brooklyn Bridge - wander over, check out the skyline Central Park - relax Top of the Rock - watch the transition from day to night Ground Zero - must go and pay respect. Times Square - sense the hustle and bustle Staton Island Ferry - relax a little Statue of Liberty - view from the ferry? Empire State Building - must do! Grand Central Station - pass through and see the architecture Ellis Island - not sure about this? Carnegie Deli - take in a cheesecake. Have we missed anything? Given we are a party of 8 will we need to book restaurants? any suggestions nearby the hotel offering good steaks and pizza (sorry I know this is a very subjective question). Thanks very much in advance - counting down the days. N16E
SummerSh...:	Wow, that's a lot to try and squeeze into such a short visit! I'd eliminate Ellis Island and Macy's. Ellis Island just takes up too much time, and Macy's – even though it's the original and the ẅorld's largest store¨- is just a dept. store, you can find a branch in Florida too. You shouldn't have to make dinner reservations if you avoid the pre-theater crowd in the area. Just go at 8:00. The best pizza is at John's, on W. 44th (four blocks from your hotel). It's in a converted church, so very interesting. I will advise you to make reservations NOW for Carmine's. Even though the place is enormous, it's really tough to get a reservation. That's where I'd suggest you get your steak fix. Their Flintstone-sized Porterhouse is yummy! Everything in Carmine's is served family style, so you'll only need two steaks for your entire group. And one big bowl of pasta will feed everyone. If you go to the World Trade Center site, please also visit St. Paul's Chapel, it's just a block away: www.saintpaulschapel.org And don't go to the Carnegie JUST for the cheesecake (even though it's fantastic!). You've gotta try their pastrami!
N16E:	Thanks Bettina, What order do you think we should visit these places in? Took a look at Carmines - the pictures on the website make it look a little formal? Thanks again. N16E.
	Summaries
HMNet:	traveling over to new york on the route to florida on wednesday 28th march. he is hoping to use the subway to take the belvedere . they are two families of 4 -4 adults and they should visit around 8 am . is also looking for advice on what to do on which subway lines to take and their hotel is fantastic . mentions that they are a party of 8 in advance and they will need to book down ellis island . they also recommend going to carnegie deli and take pizza in carnegie deli . recommends to make reservations for 44th.
LED$_{large}$:	N16E is traveling to New York on Wednesday 28th March, flying out on Saturday 31st, gives them around 2 and a half days to see the city. They are two families of 4 - 4 adults 4 kids, aged 7 to 13. They have a list of places they are looking to visit/see. They would like to visit ESB first thing around 8am and TOTR around dusk. They were hoping to use the subway to get around and their hotel is The Belvedere just off 8th Ave on 48th street. They ask which order should they visit these sights? In which order? Any info on which subway lines to take would be fantastic. They also ask about steaks and pizza nearby the hotel offering good steaks. SummerShowers says to eliminate Ellis Island and Macy's. Ellis Island just takes up too much time, and Macy's – even though it's the original.
Bart$_{base}$:	N16E needed help to plan for a two day trip to New York. N16E was traveling with his family of four and had planned to visit the Empire State Building, Central Park, Top of the Rock, Times Square, Statue of Liberty, Grand Central Station, Ellis Island, Carnegie Deli, St. Paul's Chapel, Macy's, and John's all in the same short time frame. He/She also wanted to know the order of things to do in the city, and how to get around. Bettina said that they should make reservations NOW for Carmine's.
TANET:	N16E will be traveling to New York and needs help to plan for a two day stopover. They will be staying at the Belvedere Hotel, and would like to visit the Top of the Rock, and the Empire State Building. They are two families of four and will be going to Macy's. They plan on going to Ground Zero, Central Park, Times Square, Statue of Libery, and Grand Central Station. They also ask about dinner reservations, and suggestions on places that offer good steak and pizza. SummerShowers mentions that John's has a lot of steak and may be a little too formal. He/She suggests Ellis Island, and to avoid the pre-theater crowd. Paul's and Carmine's are two family of four. N16E asks about a cheesecake in the Carnegie Deli. Summershowers responds that Carmine's is a good option as well and recommended a place called Le Pain de St. Paul's. For steak Carmine, go to Paul's Chapel and get pastrami for cheesecake.
Ground Truth:	N16E needed help to plan for a two day stopover in NYC en route to Florida, and wanted opinion on his/her itinerary. N16E said that they were staying at the Belvedere, and wanted to start at the Empire State Building and end at the Top of the Rock. N16E said that they had a party of 8 and that the Empire State Building was a must stop. N16E planned to browse at Macy's, stroll down the Brooklyn Bridge, go to Central Park to relax, visit the Top of the Rock, and see Ground Zero on the first day. On the second day N16E said that they would go to Times Square, ride the Staten Island Ferry, and see the Statue of Libery. N16E wanted to know if they could fit in Ellis Island and a visit to Grand Central Station, and still get a cheesecake for Carnegie Deli. Summershowers responded saying that was a lot to do in two days, and recommended that they drop Macy's and Ellis Island. Summershowers said that John's had the best pizza and recommended that N16E get reservations for Carmine's immediately. Summershowers said that Carmine's was family style and recommended the Flintstone-sized Porterhouse steak. Summershowers said that if N16E goes to the World Trade Center, to stop by St. Paul's Chapel, and to get pastrami a Carnegie Deli as well as a cheesecake. N16E thanked Bettine, and asked for a recommended order in visiting places, and said the Carmine's might be a little too formal.

Table 8: A case from FORUM. The conversation's domain is trip. We underline some vital facts in the conversation. Red denotes incorrect content in the generated summaries. Blue indicates what appears in TANET's summary but is not covered by the ground truth.