Improving Abstractive Dialogue Summarization with Graph Structures and Topic Words

Lulu Zhao
Beijing University of Posts and Telecommunications, Beijing, China
zhaoll@bupt.edu.cn

Weiran Xu *
Beijing University of Posts and Telecommunications, Beijing, China
xuweiran@bupt.edu.cn

Jun Guo
Beijing University of Posts and Telecommunications, Beijing, China
guojun@bupt.edu.cn

Abstract

Recently, people have been beginning paying more attention to the abstractive dialogue summarization task. Since the information flows are exchanged between at least two interlocutors and key elements about a certain event are often spanned across multiple utterances, it is necessary for researchers to explore the inherent relations and structures of dialogue contents. However, the existing approaches often process the dialogue with sequence-based models, which are hard to capture long-distance inter-sentence relations. In this paper, we propose a Topic-word Guided Dialogue Graph Attention (TGDGA) network to model the dialogue as an interaction graph according to the topic word information. A masked graph self-attention mechanism is used to integrate cross-sentence information flows and focus more on the related utterances, which makes it better to understand the dialogue. Moreover, the topic word features are introduced to assist the decoding process. We evaluate our model on the SAMSum Corpus and Automobile Master Corpus. The experimental results show that our method outperforms most of the baselines.

1 Introduction

Due to the explosive growth of the textual information, text summarization, which is an important task in Natural Language Processing (NLP), has been widely studied for several years. It can be categorized into two types: extractive and abstractive. Extractive methods select sentences or phrases from the source text directly (Nallapati et al., 2017; Zhou et al., 2018; Zhang et al., 2018a; Wang et al., 2019), while abstractive methods, which are more similar to how humans summarize texts, attempt to understand the semantic information of source text and generate new expressions as the summary. Recently, neural network methods have led to encouraging results in the abstractive summarization of single-speaker documents like news, scientific publications, etc (Rush et al., 2015; Gehrmann et al., 2018; Xu et al., 2020). These approaches employ a sequence-to-sequence general framework where the documents are fed into an encoder network and another decoder network learns to decode the summary.

With the popularity of phone calls, e-mails, and social network applications, people share information in more different ways, which are often in the form of dialogues. Different from news texts, dialogue is a dynamic information exchange flow, which is often informal, verbose and repetitive, sprinkled with false-starts, backchanneling, reconfirmations, hesitations, and speaker interruptions (Sacks et al., 1974). Besides, utterances are often turned from different interlocutors, which leads to the topic drifts, and lower information density. These problems need to be solved using natural language generation techniques with a high level of semantic understanding.

Some early works benchmarked the abstractive dialogue summarization task using the AMI meeting corpus, which contains a wide range of annotations, including dialogue acts, topic descriptions, etc. (Carletta et al., 2005; Mehdad et al., 2014; Banerjee et al., 2015). Goo and Chen (2018) proposed to use the high-level topic descriptions (e.g. costing evaluation of project process) as the gold references and leveraged dialogue act signals in a neural summarization model. They assumed that dialogue acts

∗ Corresponding author

Proceedings of the 28th International Conference on Computational Linguistics, pages 437–449
Barcelona, Spain (Online), December 8-13, 2020

indicated interactive signals and used these information for a better performance. Because this meeting dataset has a low number of summaries and is different from real dialogues, this network can not reflect its effectiveness on dialogue summarization. Customer service interaction is also a common form of dialogue, which contains questions of the user and solutions of the agent. Liu et al. (2019a) collected a dialogue-summary dataset from the logs in the DiDi customer service center. They proposed a novel Leader-Writer network, which relies on auxiliary key point sequences to ensure the logic and integrity of dialogue summaries, and designs a hierarchical decoder. The rules of labeling the key point sequences are given by domain experts, which needs to consume a lot of human efforts. Considering the lack of high-quality datasets, Gliwa et al. (2019) created the SAMSum Corpus and further investigated the problems of dialogue summary generation. They only proposed the dataset and experimented with general networks of text summarization. Although some progress has been made in abstractive dialogue summarization task, previous methods do not develop specially designed solutions for dialogues, and are all dependent on sequence-to-sequence models, which can not handle the sentence-level long-distance dependency and capture the cross-sentence relations.

To mitigate these issues, an intuitive way is to model the relations of sentences using the graph structures, which can break the sequential positions of dialogues and directly connect the related long-distance utterances. In this paper, we propose a Topic-word Guided Dialogue Graph Attention (TGDGA) network that discovers the intra-sentence and inter-sentence relations by graph neural networks, and generates summaries relied on the graph-to-sequence framework and topic words. Nodes of different granularity levels represent topic word features and utterance sequence features, respectively. The edges in the graph are initialized by the linguistic information relationships between the nodes. The masking mechanism operated in the graph self-attention layer only leverages related utterances and filters out redundant utterances. The dialogue graph aggregates the useful conversation history and captures cross-sentence relations effectively. Besides, we encode the topic words to the topic information representation and integrate it into the decoder, to guide the process of generation.

The key contributions of this work include:

- To the best of our knowledge, we are the first to construct the whole dialogue as a graph for abstractive dialogue summarization. The proper graph structure permits easier analysis of various key information in the dialogue and separates available utterances. Graph neural networks avoid the problem of long-distance dependency and the cross-sentence relations can be extracted, which makes the information flow of the dialogue more clearer.

- We devise a topic-word guided graph-to-sequence network that generates dialogue summaries in an end-to-end way. The topic word information is leveraged through graph attention mechanism, coverage mechanism, and pointer mechanism, which makes the summary more centralized with key elements. Experiments show that our model outperforms all baselines on two benchmark datasets without the pre-trained language models.

2 Related Work

2.1 Abstractive document summarization

With the development of the encoder-decoder framework on machine translation, more and more researchers take note of its great potential in document summarization area, especially for abstractive methods. Rush et al. (2015) were the first to apply the general seq2seq model with an attention mechanism. Li et al. (2017) creatively incorporated the variational auto-encoder into the seq2seq model to learn the latent structure information. To alleviate the Out-Of-Vocabulary (OOV) problem, Gu et al. (2016) introduced the copy mechanism in sequence-to-sequence learning by copying words from the source text. See et al. (2017) proposed a pointer-generator network and incorporated an additional coverage mechanism into the decoder. Moreover, Reinforcement Learning (RL) approaches have been proved to further improve the performance. Sharma et al. (2019) presented a two-step approach: an entity-aware content selection module to identify salient sentences from the input and a generation module to generate summaries. Reinforcement learning was used to connected the two components.

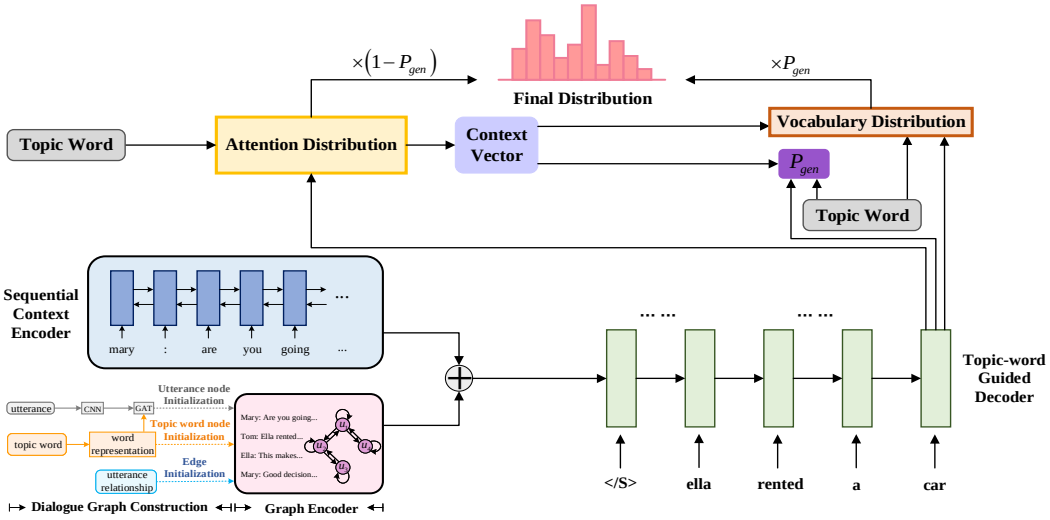

Figure 1: A general architecture of Topic-word Guided Dialogue Graph Attention model for abstractive dialogue summarization.

2.2 Abstractive dialogue summarization

Due to the lack of publicly available resources, some tentative works for dialogue summarization have been carried out in various fields. Goo and Chen (2018) produced the summaries of AMI meeting corpus based on the annotated topics the speakers discuss about. In their work, a sentence-gated mechanism was used to jointly model the explicit relationships between dialogue acts and summaries. For customer service, Liu et al. (2019a) proposed a model where a key point sequence acts as an auxiliary label in the training procedure. In the prediction procedure, the Leader-Writer network predicts the key point sequence first and then uses it to guide the prediction of the summaries. For Argumentative Dialogue Summary Corpus, Ganesh and Dingliwal (2019) used the sequence tagging of utterances for identifying the discourse relations of the dialogue and fed these relations into an attention-based pointer network. From consultation between nurses and patients, Liu et al. (2019b) arranged a pilot dataset. They presented an architecture that integrates the topic-level attention mechanism in the pointer-generator network, utilizing the hierarchical structure of dialogues. Besides, Gliwa et al. (2019) introduced a new abstractive dialogue summarization dataset and verify the performances of general sequence-based models.

2.3 Graph Neural Networks for NLP

The Graph Neural Networks (GNNs) have attracted growing attention recently, which are good for representing graph structures in NLP tasks, such as sequence labeling (Marcheggiani and Titov, 2017), relation classification (Zhao et al., 2020), text classification (Zhang et al., 2018b), and text generation (Song et al., 2018). For summary task, early traditional works made use of inter-sentence cosine similarity to build the connectivity graph like LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004). Later, some works used discourse inter-sentential relationships to build the Approximate Discourse Graph (ADG) (Yasunaga et al., 2017) and Rhetorical Structure Theory (RST) graph (Xu et al., 2019). They usually rely on external tools and cause error propagation. To avoid these problems, Transformer encoder was used to create a fully-connected graph that learns relations between pairwise sentences (Zhong et al., 2019). Nevertheless, how to construct an effective graph structure for summarization remains a difficult problem.

3 Methodology

In this section, we introduce the Topic-word Guided Dialogue Graph Attention Network for the summary generation. The TGDGA includes four parts: (1) Dialogue Graph Construction (2) Graph Encoder (3) Sequential Context Encoder (4) Topic-word Guided Decoder. Figure 1 presents the overview of our model.

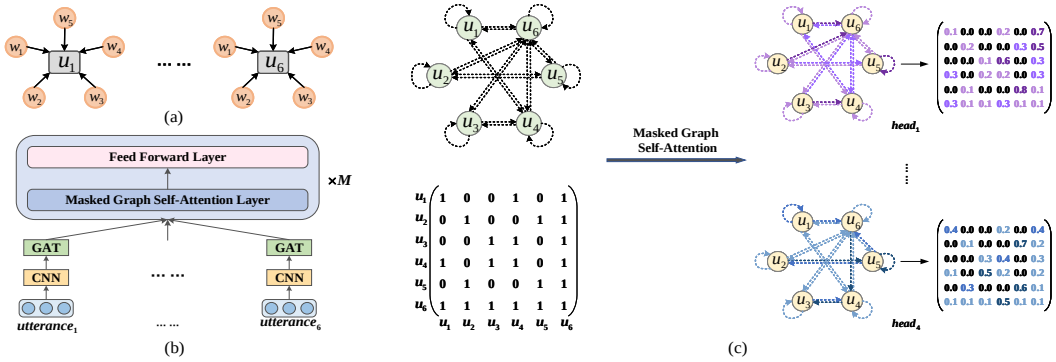

Figure 2: (a) The detailed illustration of graph attention mechanism between utterance node and topic word node. Grey boxes and orange circles represent utterance nodes and topic word nodes respectively. (b) The general architecture of graph encoder. (c) Masked graph self-attention mechanism transforms the original dialogue into different edge-weighted graphs. Numbers in the matrix denote the weights.

3.1 Dialogue Graph Construction

Given an input graph $G = \{V, E\}$, V stands for a node set which is defined as $V = V_u \cup V_w$, and E is the edge set which is defined $E = E_{uu} \cup E_{uw}$. Here, $V_u = \{u_1, ..., u_m\}$ and $V_w = \{w_1, ..., w_n\}$ donate m utterances and n topic words of the dialogue, respectively. $e_{ij}^{uu} \in E_{uu}$ $(i \in \{1, ..., m\}, j \in \{1, ..., m\})$ corresponds to the relationship between utterance nodes. $e_i \in E_{uw}$ $(i \in \{1, ..., n\})$ represents the relationship between utterance nodes and topic word nodes. The nodes and edges in the graph are initialized in the following way.

Node initialization Considering that the topic word information plays an important role in the dialogue, we assign some high probable topic words trained by LDA model (Hoffman et al., 2010). LDA is a probabilistic topic model and its parameters are estimated using the collapsed Gibbs sampling algorithm (Zhao et al., 2011). Moreover, the names of all interlocutors mentioned in the dialogue history are also added into the topic word set. We use a Convolutional Neural Network (CNN) with different filter sizes to capture the local feature representations z_i for each utterance u_i. Each topic word w_i is transformed into a real-valued vector representation ε_i by looking up the word embedding matrix, which is initialized by a random process. To update utterance node representations, we introduce a shared graph attention mechanism (Veličković et al., 2018) which can characterize the strength of contextual correlations between utterance node u_j and topic word node w_i $(i \in \mathcal{N}_j)$, where $\mathcal{N}_j$ is the topic neighborhood of utterance node u_j, as shown in Figure 2 (a). Besides, it can also diminish the repercussion of impertinent topic words and emphasize the relevant ones to the utterance. The utterance node representation x_j is calculated as follows:

$$
\begin{aligned}
A_{ij} &= f\left(\varepsilon_i, z_j\right) = \varepsilon_i^T z_j \\
\alpha_{ij} &= softmax_i\left(A_{ij}\right) = \frac{exp\left(A_{ij}\right)}{\sum_{k \in N_j} exp\left(A_{kj}\right)} \\
x_j &= \sigma\left(\sum_{i \in N_j} \alpha_{ij} W_a \varepsilon_i\right)
\end{aligned}
\tag{1}
$$

where W_a is a trainable weight and α_{ij} is the attention coefficient between ε_i and z_j.

Edge initialization If we hypothesize that each utterance node is contextually dependent on all the other nodes in a dialogue, then a fully connected graph would be constructed. However, this leads to a huge amount of computation. Therefore, we adopt a strategy to construct the edges of the graph, which associates the utterances of the dialogue according to the topic word information. If node u_i and u_j share at least one topic word, an edge $e_{ij}^{uu} = 1$ is assigned to them.

3.2 Graph Encoder

After we get the constructed graph G with utterance node features x and the edge set E, we feed them into a graph encoder to represent the dialogue. As shown in Figure 2 (b), the graph encoder is composed

of M identical blocks and each block consists of two types of layers: the masked graph self-attention layer, and the feed forward layer.

Masked Graph Self-Attention Layer Different parts of the dialogue history have distinct levels of importance that may influence the summary generation process. We choose to use the masked attention mechanism to focus more on the salient utterances. The general self-attention operation captures the interactions between two arbitrary positions of a single sequence (Vaswani et al., 2017). However, our masked self-attention operation only calculates the similarity relationship between two connected nodes in the graph and masks the irrelevant edges, as shown in Figure 2 (c). The similarity relationship is regarded as the edge weight which can be learned by the model in an end-to-end fashion. This layer is designed as follows:

$$\begin{aligned} head_i^l &= Attention\left(QW_i^{Q,l}, KW_i^{K,l}, VW_i^{V,l}\right) \\ Attention\left(Q,K,V\right) &= softmax\left(\frac{Q\times K}{\sqrt{d}}\right)V \\ g^l &= \left[head_1^l;...;head_H^l\right]W^{o,l} \end{aligned} \tag{2}$$

where W^o, W_i^Q, W_i^V, and W_i^K are weight matrices, H is the head number, and d is the dimension of utterance node features $x \in \mathbb{R}^{m\times d}$. In the first block, Q, K, and V are x. For the following blocks l, they are the feed forward layer output vector $f^{l-1} \in \mathbb{R}^{m\times d}$ of block $l-1$.

Feed Forward Layer This layer contains two linear combinations with a $ReLU$ activation in between just as Transformer (Vaswani et al., 2017). Formally, the output of the linear transformation layer is defined as:

$$f^l = ReLU\left(g^l w_1^l + b_1^l\right)w_2^l + b_2^l \tag{3}$$

where w_1, and w_2 are weight matrices. b_1, and b_2 are bias vectors.

3.3 Sequential Context Encoder

Because dialogues are sequential by nature, parts of the contextual information will also flow along the sequence. The tokens of the dialogue are fed one-by-one into a single-layer bidirectional LSTM unit, producing a sequence of encoder hidden states $h_i, i = 1, 2, ..., N$. Finally, we concatenate the last layer representation of graph encoder f^M and the last state representation of the sequential context encoder h_N as the initial state of the decoder s_0.

$$s_0 = \left[f^M; h_N\right] \tag{4}$$

3.4 Topic-word Guided Decoder

Most encoder-decoder models just use the source text as input, which leads to a lack of topic word information in the generated summaries. We propose a topic-word guided decoder to enhance the topic word information from two aspects: the coverage mechanism and pointer mechanism. In detail, we take mean pooling over all topic word node representations of a dialogue as the topic information representation $\bar{\varepsilon}$, representing the prior knowledge in the decoding steps:

$$\bar{\varepsilon} = \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i \tag{5}$$

Coverage mechanism Repetition is a common problem in the generation task, especially the names of interlocutors, and important actions. For instance, "Lilly and Lilly are going to eat salmon". Therefore, we adapt the coverage mechanism to solve the problem. Traditional coverage mechanism is hard to identify topic word information, which just involves the decoder state and the encoder hidden states (See et al., 2017). We add the topic words into the coverage mechanism:

$$a^t = softmax\left(v^T tanh\left(W_h h_i + W_s s_t + W_c c_i^t + W_k\bar{\varepsilon} + b_{attn}\right)\right) \tag{6}$$

where $c^t = \sum_{t'=0}^{t-1} a^t$. v, W_h, W_s, W_c, W_k, and b_{attn} are learnable parameters. The coverage vector c^t makes it easier for the attention mechanism to avoid repeatedly attending to the same locations, and thus avoids generating repetitive text. The attention distribution is used to produce a weighted sum of encoder hidden states, known as the context vector h_t^*:

$$h_t^* = \sum_i a_i^t h_i \tag{7}$$

To produce the vocabulary distribution P_{vocab}, the context vector, decoder state, and the topic vector are fed through two linear layers:

$$P_{vocab} = softmax\left(U'\left(U\left[s_t, h_t^*, \bar{\varepsilon}\right] + b\right) + b'\right) \tag{8}$$

where U, U', b and b' are learnable parameters.

Pointer mechanism Due to the limitation of the fixed vocabulary size, some topic word information may be lost in the summaries. Therefore, we modify the pointer mechanism which can extend the target vocabulary to include topic words. The topic vector $\bar{\varepsilon}$, the context vector h_t^*, the decoder input d_t, and the decoder hidden state s_t are taken as inputs to calculate a soft switch p_{gen}, which is used to choose between generating a word from the target vocabulary or copying a word from the input text:

$$p_{gen} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_d^T d_t + w_k^T \bar{\varepsilon} + b_{gen}) \tag{9}$$

where $w_{h^*}^T$, w_s^T, w_d^T, w_k^T, and b_{gen} are learnable parameters. σ is the sigmoid function. We obtain the following probability distribution over the extended vocabulary:

$$P\left(w\right) = p_{gen} P_{vocab}\left(w\right) + \left(1 - p_{gen}\right) \sum_{i:w_i=w} a_i^t \tag{10}$$

Note that if w is an out-of-vocabulary word, $P\left(w\right)$ is zero.

3.5 Loss Function

For each timestep t, the loss function consists of the negative log likelihood loss of the target word w_t^* and the coverage loss . The composite loss function is defined as:

$$loss_t = -logP\left(w_t^*\right) + \lambda \sum_i min\left(a_i^t, c_i^t\right) \tag{11}$$

4 Dataset and Experimental Setup

4.1 Dataset

We perform our experiments on the SAMSum Corpus and the Automobile Master Corpus, which are both new corpora for dialogue summarization. The SAMSum Corpus is an English dataset about natural conversations in various scenes of the real-life, which includes chit-chats, gossiping about friends, arranging meetings, discussing politics, consulting university assignments with colleagues, etc (Gliwa et al., 2019). The standard dataset is split into 14732, 818, and 819 examples for training, development, and test. The Automobile Master Corpus is from the customer service question and answer scenarios. [1] We use a portion of the corpus that consists of high-quality text data, excluding picture and speech data. It is split into 183460, 1000, and 1000 for training, development, and test. More statistics of two datasets are in the Table 1.

[1]This dataset is released by the AI industry application competition of Baidu.

Instance	SAMSum Corpus				Automobile Master Corpus			
	Avg_Dia	Avg_Tur	Avg_Sum	Avg_Tw	Avg_Dia	Avg_Tur	Avg_Sum	Avg_Tw
Train	120.26	11.13	22.81	13.74	181.94	11.18	23.25	14. 50
Dev	117.46	10.72	22.80	13.69	190.03	10.25	22.82	14.36
Test	122.71	11.24	22.47	13.80	181.13	11.21	22.56	14.57

Table 1: Data statistics. Avg_Dia, and Avg_Sum are the average number of tokens in dialogues and summaries, respectively. Avg_Tur is average number of utterances in dialogues, and Avg_Tw is the average number of topic word extracted from dialogues.

4.2 Training details

We filter stop words and punctuations from the training set to generate a limited vocabulary size of 40k. The dialogues and summaries are truncated to 500, and 50 tokens, and we limit the length of each utterance to 20 tokens. The embedding size is set to 128. The word embeddings are shared between the encoder and the decoder. The hidden size of graph encoder and sequential context encoder is 128 and 256, respectively. We use a block number of 2, and the head number of 4 for masked graph self-attention operation. At test time, the minimum length of the generated summary is set to 15, and the beam size is 5. For all the models, we train for 30000 iterations using Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.001 and the batch size of 8.

4.3 Baseline methods

We compare our proposed model with the following baselines:

Longest-3: This model is commonly used in the news summarization task, which treats 3 longest utterances in order of length as a summary.

Seq2Seq+Attention: This model is proposed by Rush et al. (2015), which uses an attention-based encoder that learns a latent soft alignment over the input text to help inform the summary.

Transformer: This model is proposed by Vaswani et al. (2017), which relies entirely on an attention mechanism to draw global dependencies between the input and output.

LightConv: This model is proposed by Wu et al. (2019), which has a very small parameter footprint and the kernel does not change over time-steps.

DynamicConv: This model is also proposed by Wu et al. (2019), which predicts a different convolution kernel at every time-step and the dynamic weights are a function of the current time-step only rather than the entire context.

Pointer Generator: This model is proposed by See et al. (2017), which aids the accurate reproduction of information by pointing and retains the ability to produce new words through the generator.

Fast Abs RL: This model is proposed by Chen and Bansal (2018), which constructs a hybrid extractive-abstractive architecture, with the policy-based reinforcement learning to bridge together the two networks.

Fast Abs RL Enhanced: This model is a variant of Fast Abs RL, which adds the names of all other interlocutors at the end of utterances.

5 Results and Discussions

5.1 Main Results

Results on SAMSum Corpus The results of the baselines and our model on SAMSum dataset are shown in Table 2. We evaluate our models with the standard ROUGE metric, reporting the F1 scores for ROUGE-1, ROUGE-2, and ROUGE-L (which respectively measure the word-overlap, bigram-overlap, and longest common sequence between the reference summary and the summary to be evaluated). By observation, the inclusion of a Separator [2] is advantageous for most models, because it improves the discourse structure. Compared to the best performing model Fast Abs RL Enhanced, the TGDGA model

[2]Separator is a special token added artificially, e.g. <EOU> for models using word embeddings, | for models using subword embeddings. The use of it is proposed by Gliwa et al. (2019).

obtains 1.16, 1.09, and 1.26 points higher than it for R-1, R-2, and R-L. The masked graph self-attention operation of our model and the extractive method of Fast Abs RL Enhanced model play a similar role in filtering important contents in dialogues. However, our model does not need to use reinforcement learning strategies, which greatly simplifies the training process. Besides, the TGDGA model outperforms the Transformer model based on fully connected relationships, which demonstrates that our dialogue graph structures effectively prune unnecessary connections between utterances. Since the additional topic word information, our model also surpasses the pointer generator model by 2.23, 3.87, and 3.86 points.

Results on Automobile Master Corpus Table 2 shows experimental results on Automobile Master dataset. Our TGDGA achieves Rouge-1, Rouge-2, and Rouge-L of 42.98, 17.58, and 38.11, which outperforms the baseline methods by different margins. Unlike the SAMSum dataset, Fast Abs RL Enhanced model has no obvious advantage over other sequence models. This is because that the average number of utterances in the dialogue is more and the information is more scattered. We also notice that our model outperforms the pointer generator model as well. Due to the limited computational resource, we don't apply a pre-trained contextualized encoder (i.e. BERT) to our model, which we will regard as our future work. Therefore, we only compare with models without BERT for the sake of fairness.

Model	**SAMSum Corpus**			**Automobile Master Corpus**		
	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
Longest-3	32.46	10.27	29.92	30.72	9.07	28.14
Seq2Seq	21.51	10.83	20.38	25.84	13.82	25.46
Seq2Seq + Attention	29.35	15.90	28.16	30.18	16.52	29.37
Transformer	36.62	11.18	33.06	36.21	11.13	34.08
Transformer + Separator	37.27	10.76	32.73	37.43	11.87	34.97
LightConv	33.19	11.14	30.34	34.68	12.41	31.62
DynamicConv	33.79	11.19	30.41	34.72	12.45	31.86
DynamicConv + Separator	33.69	10.88	30.93	34.41	12.38	31.22
Pointer Generator	38.55	14.14	34.85	39.17	15.39	34.76
Pointer Generator + Separator	40.88	15.28	36.63	39.23	15.42	34.53
Fast Abs RL	40.96	17.18	39.05	39.82	15.86	36.03
Fast Abs RL Enhanced	41.95	18.06	39.23	40.13	16.17	36.42
TGDGA (ours)	43.11	19.15	40.49	42.98	17.58	38.11

Table 2: Results in terms of Rouge-1, Rouge-2, and Rouge-L on the SAMSum Corpus test set and Automobile Master Corpus test set.

Human Evaluation We further conduct a manual evaluation to assess the models. Since the ROUGE score often fails to quantify the machine generated summaries (Schluter, 2017), we focus on evaluating the relevance and readability of each summary. Relevance is a measure of how much salient information the summary contains, and readability is a measure of how fluent and grammatical the summary is. 50 samples are randomly selected from the test set of SAMSum Corpus and Automobile Master Corpus, respectively. The reference summaries, together with the dialogues are shuffled then assigned to 5 human annotators to score the generated summaries. Each perspective is assessed with a score from 1 (worst) to 5 (best) to indicate whether the summary is understandable and gives a brief overview of the text. The average score is reported in Table 3. As we can see, Pointer Generator suffers from repetition and generates many trivial facts. For Fast Abs RL Enhanced model, it successfully concentrates on the salient information, however, the dialogue structure is not well constructed. By introducing the topic word information and coverage mechanism, our TGDGA model avoids repetitive problems and better extracts the core information in the dialogue.

5.2 Ablation Study

We examine the contributions of three main components, namely, graph encoder, topic information in coverage mechanism, topic information in pointer mechanism, using the best-performing TGDGA model on test set of the SAMSum corpus. The results are shown in Table 4. First, we discuss the effect of

Dataset	Model	Relevance	Readability
SAMSum	Pointer Generator + Separator	2.36	4.25
	Fast Abs RL Enhanced	2.67	4.73
	TGDGA (ours)	2.91	4.86
Automobile Master	Pointer Generator + Separator	2.41	4.18
	Fast Abs RL Enhanced	2.59	4.35
	TGDGA (ours)	2.88	4.62

Table 3: Human Evaluation on the SAMSum Corpus test set and Automobile Master Corpus test set.

the graph encoder. The removal of it (i.e. TGDGA w/o GE) leads performance to drop greatly. It suggests that graph encoder effectively uses the conversational structure to capture utterance-level long-distance dependencies. Moreover, after we get rid of the topic information in coverage mechanism (i.e. TGDGA w/o TICM) and in pointer mechanism (i.e. TGDGA w/o TIPM), respectively, both of the models could not keep as competitive as TGDGA, verifying that topic information is significant for generating informative and faithful summaries.

Model	Rouge-1	Rouge-2	Rouge-L
TGDGA	43.11	19.15	40.49
TGDGA w/o GE	42.87	18.35	39.71
TGDGA w/o TICM	43.05	19.10	40.37
TGDGA w/o TIPM	42.93	18.97	40.14

Table 4: An ablation study for three components in TGDGA on test set of SAMSum.

Dialogue	Lilly: sorry, I'm gonna be late. Lilly: don't wait for me and order the food. Gabriel: no problem, shall we also order something for you? Gabriel: so that you get it as soon as you get to us? Lilly: good idea! Lilly: pasta with salmon and basil is always very tasty there.
Reference	Lilly will be late. Gabriel will order pasta with salmon and basil for her.
Longest-3	gabriel: no problem, shall we also order something for you? gabriel: so that you get it as soon as you get to us? lilly: pasta with salmon and basil is always very tasty there.
Pointer Generator	lilly will be late. lilly will order pasta.
Fast Abs RL Enhanced	lilly will be late. lilly and gabriel are going to pasta with salmon and basil is always tasty.
TGDGA (ours)	lilly will be late. she wants gabriel to order pasta with salmon and basil.

Table 5: An example of summaries generated by different models. The pink font represents the topic word and the blue font represents interlocutor' name.

5.3 Case Study

Table 5 shows an example of dialogue summaries generated by different models. The summary generated by the Pointer Generator model repeats the same name "lilly" and only focuses some pieces of information in the dialogue. For Fast Abs RL Enhanced model, it adds information about the other interlocutors, which makes the generated summary contain both interlocutors' names: lilly and gabriel, and obtains other valid key elements, e.g. pasta with salmon and basil because of the extractive method. However, Fast Abs RL Enhanced model usually makes a mistake in deciding who performs the action (the subject) and who receives the action (the object), which may be due to the way the dialogue is constructed. Important utterances are firstly chose and then summarizes each of them separately. This leads to the narrowing of the context and losing pieces of important information. Our model uses topic

word information to guide the construction of dialogue structure, and on the basis of not deleting the dialogue content, we use the masked graph self-attention mechanism to strengthen the expression of the main content in the dialogue. Topic words are also used in the decoding process to match person names and events correctly.

5.4 Attention Visualization

Intuitively, masked graph self-attention mechanism models the interaction between contextual utterances with relevance. If the model works as expected, more attention should be paid to utterances with similar topic word information. To further analyze the attention learned in the model, we visualize the utterance attention weights when constructing dialogue graph structures in Figure 3. The figure is colored with different levels of attention, in which the white one represents that there is no attention weight between two utterances, and the darker one represents that there is a greater attention value between two utterances. In this example, for utterance 1, utterance 6 gets the highest attention weight, and utterance 4 gets a higher weight. Utterance 2, 3, and, 5 do not participate in the attention mechanism operation at all. This suggests that in this case, the model can focus on more important utterance information correctly.

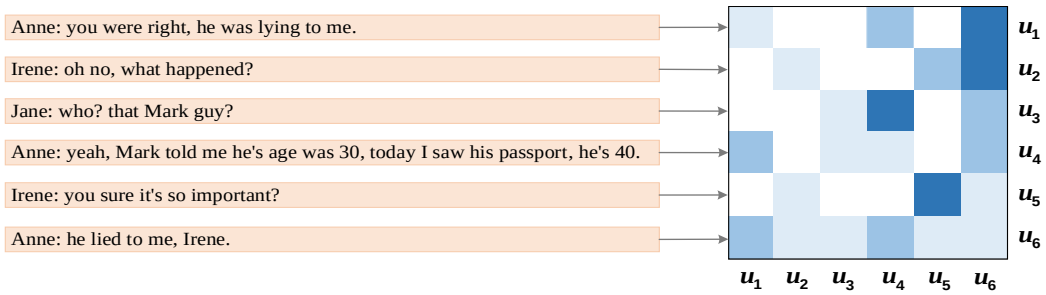

Figure 3: Visualization of attention in the TGDGA model. Darker color indicates a higher attention weight.

6 Conclusion

In this paper, we propose a Topic-word Guided Dialogue Graph Attention model to automatically generate summaries of dialogues with a graph-to-sequence framework. The dialogue is organized into an interaction graph, which improves context understanding for sentence-level long-dependency and builds more complex relations between utterances. The introduction of masking mechanism helps our model to select salient utterances and aware of the hierarchical structure of dialogues. We also incorporate topic words information into the summary generation process. Experimental results strongly support the improvements in our proposal. Furthermore, we will take the pre-trained language models into account for better encoding representations of words and expect more advanced work to be done in the area of evaluation metrics in the future.

Acknowledgments

We thank all anonymous reviewers for their constructive feedback. This work was partially supported by National Key R&D Program of China No.2019YFF0303300 and Subject II No.2019YFF0303302, MoE-CMCC “Artificial Intelligence” Project No.MCM20190701, DOCOMO Beijing Communications Laboratories Co., Ltd.

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*, WWW ’15 Companion, page 5–6, New York, NY, USA. Association for Computing Machinery.

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, page 28–39, Berlin, Heidelberg. Springer-Verlag.

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, July. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December.

Prakhar Ganesh and Saket Dingliwal. 2019. Abstractive summarization of spoken and written conversation.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium, October-November. Association for Computational Linguistics.

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, November. Association for Computational Linguistics.

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

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August. Association for Computational Linguistics.

Matthew Hoffman, Francis R. Bach, and David M. Blei. 2010. Online learning for latent dirichlet allocation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 856–864. Curran Associates, Inc.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100, Copenhagen, Denmark, September. 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 and Data Mining*, KDD '19, page 1957–1965, New York, NY, USA. Association for Computing Machinery.

Z. Liu, A. Ng, S. Lee, A. T. Aw, and N. F. Chen. 2019b. Topic-aware pointer-generator networks for summarizing spoken conversations. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 814–821.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark, September. Association for Computational Linguistics.

Yashar Mehdad, Giuseppe Carenini, and Raymond T. Ng. 2014. Abstractive summarization of spoken and written conversations based on phrasal queries. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1220–1230, Baltimore, Maryland, June. Association for Computational Linguistics.

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, July. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3075–3081. AAAI Press.

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, September. Association for Computational Linguistics.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.

Natalie Schluter. 2017. The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain, April. Association for Computational Linguistics.

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, July. Association for Computational Linguistics.

Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. 2019. An entity-driven framework for abstractive summarization. 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 3280–3291, Hong Kong, China, November. Association for Computational Linguistics.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia, July. 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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Hong Wang, Xin Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Self-supervised learning for contextualized extractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2221–2227, Florence, Italy, July. Association for Computational Linguistics.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Discourse-aware neural extractive text summarization.

Weiran Xu, Chenliang Li, Minghao Lee, and Chi Zhang. 2020. Multi-task learning for abstractive text summarization with key information guide network. *EURASIP Journal on Advances in Signal Processing*.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada, August. Association for Computational Linguistics.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018a. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 779–784, Brussels, Belgium, October-November. Association for Computational Linguistics.

Yue Zhang, Qi Liu, and Linfeng Song. 2018b. Sentence-state LSTM for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327, Melbourne, Australia, July. Association for Computational Linguistics.

Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, page 338–349, Berlin, Heidelberg. Springer-Verlag.

Lulu Zhao, Weiran Xu, Sheng Gao, and Jun Guo. 2020. Cross-sentence n-ary relation classification using lstms on graph and sequence structures. *Knowledge-Based Systems*, 207:106266.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy, July. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia, July. Association for Computational Linguistics.