Give the Truth: Incorporate Semantic Slot into Abstractive Dialogue Summarization

Lulu Zhao[1], **Weihao Zeng**[1], **Weiran Xu**[1]*, **Jun Guo**[1]
[1]Pattern Recognition & Intelligent System Laboratory
[1]Beijing University of Posts and Telecommunications, Beijing, China
{zhaoll,ZengWH,xuweiran,guojun}@bupt.edu.cn

Abstract

Abstractive dialogue summarization suffers from a lots of factual errors, which are due to scattered salient elements in the multi-speaker information interaction process. In this work, we design a heterogeneous semantic slot graph with a slot-level mask cross-attention to enhance the slot features for more correct summarization. We also propose a slot-driven beam search algorithm in the decoding process to give priority to generating salient elements in a limited length by "filling-in-the-blanks". Besides, an adversarial contrastive learning assisting the training process is introduced to alleviate the exposure bias. Experimental performance on different types of factual errors shows the effectiveness of our methods and human evaluation further verifies the results.

1 Introduction

Current state-of-the-art conditional text generation models accomplish a high level of fluency and informativeness, mostly thanks to advances in seq2seq architectures with the attention and copy mechanisms (See et al., 2017) and the pre-trained transformer-based models for natural language understanding (Lewis et al., 2019; Yang et al., 2020). Despite this progress (Kryscinski et al., 2019), there are still many limitations facing neural text summarization, the most serious of which is their tendency to generate summaries with a substantial number of factual errors. Besides, the ROUGE scores (Lin, 2004), the most commonly used evaluation metrics, are inadequate to quantify factual correctness and only capture the information coverage at token-level, i.e., n-gram overlap, which does not always convey the desired semantics and reach consensus with human judgement.

Recently, as people increasingly exchange information in the way of dialogue, giving a high-quality summarization for the dialogue is particularly necessary, which can help people quickly grasp the core information of the long dialogue history without reviewing the complex context and is significant to improve the efficiency of social contact. However, as a special kind of text form, the dialogue is usually informal and dynamic. Utterances are often said by different speakers alternately in different language styles, which leads to the description of one event being fragmented and scattered in multiple utterances. These inherent differences between dialogues and documents make it easier to product various factual errors, i.e., factual inconsistent and incomplete, in the generated summaries, as shown in Fig.1. Therefore, it is urgent to develop a neural model, focusing on exploring the factual correctness, to generate overall high-quality summaries for the multi-people dialogue scene.

There have been some recent researches on abstractive dialogue summarization, such as deploying document summarization methods to the conversation settings (Gliwa et al., 2019), utilizing the

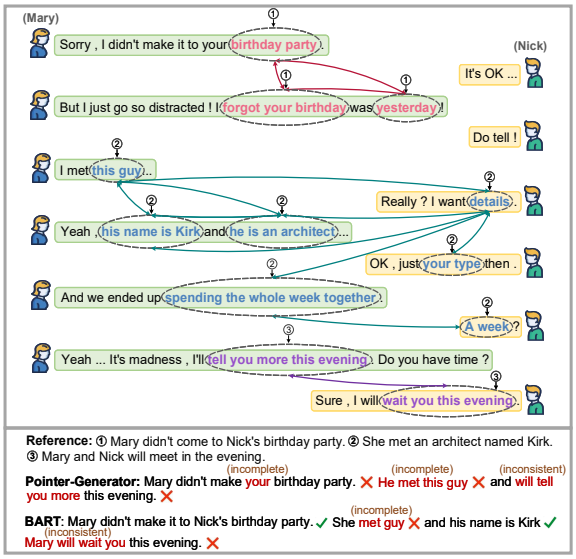

Figure 1: An example from SAMSum dataset. Dashed circles are elements of event, which are marked with same circle numbers as corresponding parts in reference. ✗ represents errors of factual inconsistent and factual incomplete.

*Weiran Xu is the corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2435–2446
November 7–11, 2021.

dialogue acts (Goo and Chen, 2018) and key point sequences (Liu et al., 2019a), topic word information (Zhao et al., 2020), and analyzing the conversational structures (Chen and Yang, 2020, 2021). Other researches have also pushed the frontier of guaranteeing the factual consistency in abstractive document summarization systems via proposing related evaluation metrics (Kryscinski et al., 2020; Maynez et al., 2020) and designing models (Dong et al., 2020; Cao et al., 2020). However, the current methods (1) fail to utilize the unique semantic and structural information of dialogues to identify the salient elements and guide the decoding process, so that to deal with factual errors for dialogue summarization, (2) lack overall evaluation metrics for factual correctness. We argue that the slot-aware structure is important to improve the performance of factual correctness for dialogue summarization. Besides, except for factual consistency metric, the factual completeness metric is also an indispensable key to evaluate factual correctness.

In this paper, we propose a Semantic Slot guided Adversarial sequence-to-sequence network (SSAnet). The SSAnet contains a heterogeneous semantic slot (HSS) graph, where different types of nodes represent different slot labels and the edges are the dependencies between slot values. Attentions of three different granularities, i.e. tokens, utterances, and slots, are unified into one architecture to promote the learning of the relationships between all granularities. Crucially, the slot-level attention mechanism can make the model directly select the appropriate slot features from the HSS graph to fill the corresponding slot in the summary sequence, which ensures the correctness and completeness of the salient information in the generated content. In the decoding process, we propose a slot-driven beam search algorithm based on Song et al. (2021) to give priority to generating salient elements in a limited length by "filling-in-the-blanks". Besides, to alleviate the exposure bias, we also use a contrastive learning strategy with adversarial perturbations (Lee et al., 2020) by actively exposing some wrong tokens during training. Finally, we propose a new evaluation metric to quantify factual completeness at the slot-level.

Our contributions can be summarized as follows: (1) To the best of our knowledge, we are the first to design a novel slot-level attention operation by copying features from an HSS graph to the corresponding slots. (2) We propose a slot-driven beam search algorithm to give priority to generating salient elements in a controlled way, which ensures the fluency and factuality of summaries. (3) A contrastive learning with adversarial perturbations is introduced to alleviate the exposure bias for dialogue summarization. (4) We perform experiments on two large-scale datasets to verify the effectiveness of our proposed methods and propose a new metric to evaluate the factual completeness.

2 Related Work

2.1 Abstractive Dialogue Summarization

Recently, abstractive dialogue summarization has attracted more attention. Some early researches adopted the dialogue act (Goo and Chen, 2018), key point sequence (Liu et al., 2019a), and topic segmentation (Liu et al., 2019b; Li et al., 2019) for dialogue summarization. However, the used datasets are either very small or non-public. Later, Gliwa et al. (2019) proposed a large-scale dataset about daily chats, named SAMSum. On this basis, some studies attempted to leverage the topic word information (Zhao et al., 2020) and the conversational structures (Chen and Yang, 2020, 2021) to improve the performance. Besides, (Zhu et al., 2021b) recently propose another large-scale media interview dataset (namely MediaSum) and evaluate several benchmark summarization models. However, current methods only focus on modeling the dialogue context via different ways to raise the ROUGE scores, but ignore whether the generated summaries are correct. To this end, we utilize the semantic slot information to guide the model to focus on generating the salient elements. While ensuring the high-level ROUGE scores, it also improves the factual consistency and completeness.

2.2 Fact-aware Summarization

When it comes to factual errors, some work focuses on designing evaluation metrics towards factual consistency, as many human evaluations have shown that ROUGE scores correlate poorly with faithfulness (Maynez et al., 2020). They range from using fact triples (Goodrich et al., 2019; Zhang et al., 2020), textual entailment predictions (Falke et al., 2019), adversarially pre-trained classifiers (Kryscinski et al., 2020), to question answering (QA) systems (Wang et al., 2020; Durmus et al., 2020). Another line of the related work focuses on enforcing factuality in summarization models. Cao et al. (2017); Zhu et al. (2021a) proposed

RNN-based and Transformer-based decoders that attend to both source texts and extracted knowledge triples, respectively. Li et al. (2018) proposed an entailment-reward augmented maximum-likelihood training objective. Dong et al. (2020); Cao et al. (2020) designed post-editing correctors to boost factual consistency in generated summaries. Our model is inherently different from these models, as we try to boost the factuality via incorporating the semantic slot information while generating the summary, instead of correcting after generating, which can significantly improve the performance of multiple factual correctness metrics without a huge drop on ROUGE scores.

3 Methodology

As illustrated in Fig.2, our methods include three parts: (1) a heterogeneous semantic slot graph, (2) a dual-encoder, and (3) a slot-aware decoder.

3.1 Heterogeneous Semantic Slot Graph

The semantic slot information is a specific concept in the dialogue system and the slots can be understood as the defined attributes of the event, that is, the backbone of the dialogue content (Yuan and Yu, 2019). Although the current neural models are supposed to, or might implicitly recognize some salient contents in dialogue, they are often difficult to describe events consistently and completely. Therefore, we extract the slot values from the dialogue context and construct a heterogeneous semantic slot graph.

Specifically, we first define the slot labels via Stanford CoreNLP and get the slot values by fine-tuning Chen et al. (2019) (See Sec 4.1). Then, we use a dependency parser tool (Manning et al., 2014) to dig out the dependencies between slot values, which are formed as $(slot_1, dependency, slot_2)$. By integrating the triples, we obtain the graph $G{=}(V, E)$, where slot values v_i are nodes in V, and nodes belonging to the same slot label are regarded as the same type of nodes. E is an adjacent matrix, where $e_{ij}{=}1$ indicates that there exists some dependency between slot values.

3.2 Dual-Encoder

To model the dialogue context, we utilize a dual-encoder, i.e., a sequence encoder and a graph encoder, which obtain the hidden representations at token-level, utterance-level, and slot-level.

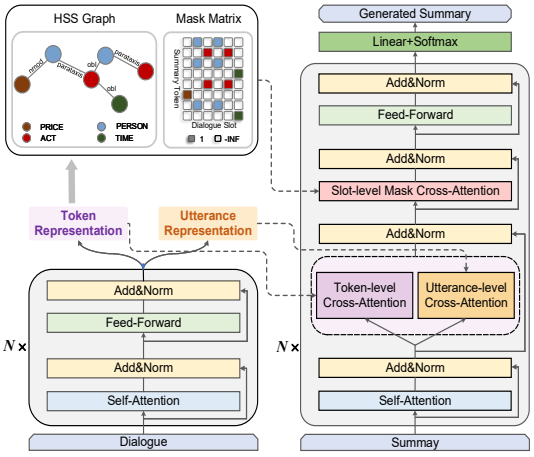

Figure 2: The structure of our proposed model SSAnet, which is based on BART.

3.2.1 Sequence Encoder

A pre-trained encoder, i.e., BART (Lewis et al., 2019), is adopted as the feature encoder to extract token representations and utterance representations due to its effectiveness in representation learning. Given a dialogue D with m utterances, $d_i{=}\{w_{i1}, ..., w_{il_i}\}$ denotes the i-th utterance with l_i tokens. we feed the input sequence into BART:

$$\{\boldsymbol{x}_{10}, \boldsymbol{x}_{11}, ..., \boldsymbol{x}_{ml_m}\} = \mathrm{BART}(\{w_{10}, w_{11}, ..., w_{ml_m}\}) \tag{1}$$

Here we add a special token $w_{i0}{=}[\mathrm{CLS}]$ $(i \in \{1, ..., m\})$ at the beginning of each utterance and regard $\boldsymbol{x}_{i0}$ as the utterance-level representation.

3.2.2 Graph Encoder

Initializers For node initialization, we employ the token-level output embeddings from sequence encoder to initialize each token in v_i and then average all token embeddings as the initial representation $\boldsymbol{s}_i$ of the node. For edge initialization, the $\mathrm{BART}(\cdot)$ is used to encode the dependency e_{ij} into the initial representation $\boldsymbol{r}_{ij}$.

Relational Graph Attention Layer Based on the constructed HSS graph, we apply a graph attention network (Veličković et al., 2018) with the dependency information to aggregate the slot-level features. This layer following a residual connection is designed as:

$$\alpha_{ij} = \frac{exp(\mathrm{LeakyReLU}(\mathbf{W}_a[\mathbf{W}_q\boldsymbol{s}_i; \mathbf{W}_k\boldsymbol{s}_j; \boldsymbol{r}_{ij}]))}{\sum_{l\in\mathcal{N}_i} exp(\mathrm{LeakyReLU}(\mathbf{W}_a[\mathbf{W}_q\boldsymbol{s}_i; \mathbf{W}_k\boldsymbol{s}_l; \boldsymbol{r}_{il}]))}$$
$$\boldsymbol{h}_i^g = \sigma(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}_v\boldsymbol{s}_i) + \boldsymbol{s}_i \tag{2}$$

where $\mathbf{W}_*$ are weight matrices, σ is the activation function, and $\mathcal{N}$ is the neighborhood of v_i in G.

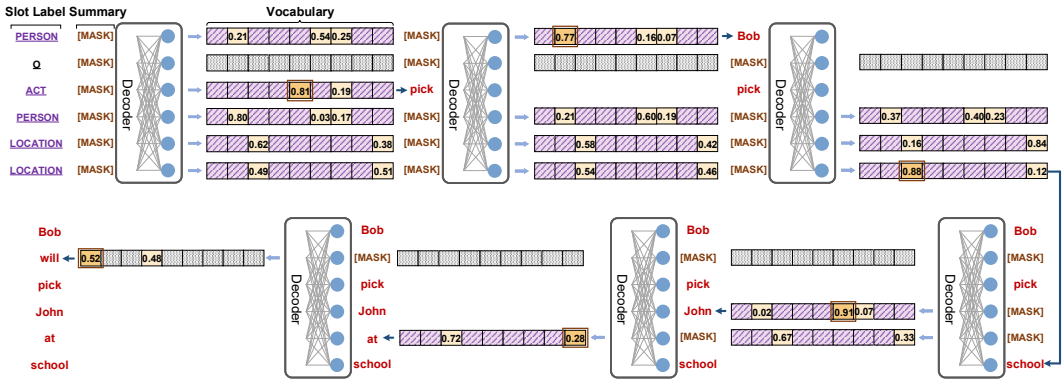

Figure 3: An illustration of the decoding process (beam_size=1). Our decoder selects the most probable token of the same slot label from all positions and give priority to non-"O" slot labels, i.e., PERSON, ACT, etc.

3.3 Slot-aware Decoder

To aggregate multi-granularity representations, we improve the BART decoder based on Transformer with two extra cross-attentions added to each decoder layer, which attends to the representations at utterance-level and slot-level. It is worth noting that the slot-level mask cross-attention is realized with a novel Mask, which represents the corresponding relationship between the slot values in HSS graph and the tokens in target summary, that is, whether they belong to the same slot label. In each decoder layer, after performing the token-level cross-attention and the utterance-level cross-attention, the slot-level mask cross-attention operation is then performed to conduct cross attentions over slot nodes $\{v_{0:|V|}\}$ of the HSS graph encoded from graph encoder to obtain the slot-attended representations.

Concretely, the summary tokens are regarded as a query matrix and the slot node representations act as a key matrix, so that every summary token simultaneously assesses how much information shall be obtained from every representation of the same type slot node. In this way, the target summary sequence representation $\mathbf{Y}$, which is the sum of token-level cross-attention representation $\mathbf{T}$ and utterance-level cross-attention representation $\mathbf{U}$, is projected to query matrix $\mathbf{Q} \in \mathbb{R}^{n \times d}$. The slot node $\mathbf{H}^g$ is projected to key matrix $\mathbf{K} \in \mathbb{R}^{|V| \times d}$ and value matrix $\mathbf{V} \in \mathbb{R}^{|V| \times d}$ by linear projections without bias: $[\mathbf{Q}; \mathbf{K}; \mathbf{V}] = \text{Linear}([\mathbf{Y}; \mathbf{H}^g; \mathbf{H}^g])$ where $[]$ is the concatenating operation. Slot-level mask cross-attention is calculated by:

$$\mathbf{H}^s = \text{softmax}(\frac{(\mathbf{Q}\mathbf{K}^{\mathrm{T}}) * M}{\sqrt{d}})\mathbf{V} \quad (3)$$

where * denotes element-wise multiplication, and $M \in \mathbb{R}^{n \times |V|}$ is the utilized mask which is defined:

$$M_{ij} = \begin{cases} 1, & label(i) = label(j) \\ -inf, & else \end{cases}$$

Just like Transformer (Vaswani et al., 2017), the output vectors are then feed into a feed-forward network for forward passing in the decoder.

Algorithm 1 Slot-driven Beam Search

procedure SLOTBEAM(DIALOGUE(D), n, L, K)
  $n \leftarrow$ max length of summary
  $L \leftarrow$ number of non-O slot value
  $\mathcal{S}^{'} \leftarrow \{[\text{MASK}] \times n\}$ Initial summary sequence
  $\mathcal{M}_d^* \leftarrow \{0, 1\}_{n \times |V|}$ Two slot-aware matrices
  $\mathcal{M}_p \leftarrow [1]_{n \times |V|}$ Position matrix
  $\mathcal{H} \leftarrow \{(0, \mathcal{S}^{'}, \mathcal{M}_d^*, \mathcal{M}_p)\}$ Hypothesis set
  for $i = 1, ..., n$ **do**
    $Cand \leftarrow \{\}$
    for $hyp \in \mathcal{H}$ **do**
      if $i \leq L$ **then**
        Calculate probability of non-**O** slot values
        $\mathcal{P}_{n \times |V|} \leftarrow \text{softmax}(\text{Gen(D}, \mathcal{S}^{'}) \odot \mathcal{M}_d^S)$
      else
        Calculate probability of **O**-slot values
        $\mathcal{P}_{n \times |V|} \leftarrow \text{softmax}(\text{Gen(D}, \mathcal{S}^{'}) \odot \mathcal{M}_d^O)$
      end if
      $score^{'}, \mathcal{S}^{'}, \mathcal{M}_p^{'}, \mathcal{M}_d^* \leftarrow hyp$
      $\mathcal{P}^{'} \leftarrow \mathcal{P} \odot \mathcal{M}_p^{'}$
      for $s_k, w_k, p_k \in \mathcal{P}^{'}$ **do**
        Record the tokens and positions
        $score^{''} \leftarrow score^{'} + s_k$
        $\mathcal{S}^{''} \leftarrow replace(\mathcal{S}^{'}, p_k, w_k)$
        $\mathcal{M}_p^{''} \leftarrow replace(\mathcal{M}_p^{'}, p_k, [0]_{1 \times |V|})$
        $Cand.add((score^{''}, \mathcal{S}^{''}, \mathcal{M}_p^{''}, \mathcal{M}_d^*))$
      end for
      $\mathcal{H} \leftarrow$ Top-K$(Cand)$
    end for
  end for
  return $\mathcal{H}$ The best summary sequence
end procedure

Slot-driven Beam Search The general beam search algorithm is a form of pruned breadth-first search and seeks the K-best candidate summaries having the highest log-likelihood to generate the next token one by one. Although beam search is one of the few NLP algorithms that has stood the test of time and has been widely used in many text generation tasks, it products a high search error rate due to the long-distance probability transition (Meister et al., 2020).

Inspired by Song et al. (2021), our slot-driven beam search simultaneously predicts the most probable tokens for all positions and decodes the summary tokens in order of priority under the guidance of semantic slot information rather than using a left-to-right order, as shown in Fig.3, which makes the most important tokens (salient elements) generated first, less important ones later. This algorithm designs two binary slot-aware matrices $\mathcal{M}_d^S$ and $\mathcal{M}_d^O$ and, both of which indicates the corresponding relations between all source tokens in the vocabulary list and the slot labels of the summary. The $\mathcal{M}_d^S$ and $\mathcal{M}_d^O$ are defined as:

$$\mathcal{M}_{d,ij}^S = \begin{cases} 1 & \{j \in slot\ value\} \cap \{label(j) = i\} \\ -inf & else \end{cases}$$

$$\mathcal{M}_{d,ij}^O = \begin{cases} 1 & \{j \notin slot\ value\} \cap \{i = O\} \\ -inf & else \end{cases}$$

Besides, a position matrix $\mathcal{M}_p \in \{0,1\}_{n \times |V|}$ is also contained to record what positions have been filled by summary tokens and what positions remain available. The detailed process is shown in Algorithm 1

3.4 Learning Objective

Following Lee et al. (2020), we introduce a contrastive learning strategy during training to improve generalization, which is realized by respectively adding a small perturbation and a large number of perturbations to the hidden representations of the target summary sequence to generate negative examples and positive examples. In this way, the conditional likelihood of the negative example is minimized but very close to the source sentence in the embedding space, and the conditional likelihood of the positive example is enforced to remain high. The overall objective is as follows:

$$\min_{\theta} -\mathcal{L}_{MLE}(\theta) + \alpha\mathcal{L}_{KL}(\theta) - \beta\{\mathcal{L}_{neg}(\theta) + \mathcal{L}_{pos}(\theta)\} \tag{4}$$

where α, β are hyperparameters, searched through cross-validation and control the importance of contrastive learning and KL divergence.

4 Experiments

4.1 Datasets

We experiment with two large-scale abstractive dialogue summarization datasets: the SAMSum dataset (Gliwa et al., 2019), which is about natural conversations in various scenes of the real-life, and the MediaSum dataset (Zhu et al., 2021b), which is about interview transcripts from NPR and CNN.

Dataset	D_tok	S_tok	D_slo	S_slo	A_tur	A_spe
SAMSum	83.9	20.3	30.5	6.4	9.9	2.2
MediaSum	1,553.7	14.4	357.5	4.2	30.0	6.5

Table 1: Data statistics. D_tok and S_tok are the token numbers of dialogues and gold summaries. D_slo and S_slo are the slot numbers of dialogues and gold summaries. A_tur and A_spe are the average numbers of turns and speakers.

Data Preprocessing We give the semantic slot information by following steps: (1) We firstly use Stanford CoreNLP (Manning et al., 2014) to do NER to get the nominal slots by integrating the high-frequency entity types with similar concepts into one slot label such as *(COUNTRY, CITY, STATE_OR_PROVINCE)→ LOCATION* and retaining the low-frequency entity types with special significance such as *MONEY→PRICE*. (2) We then use Stanford CoreNLP to do Pos Tagging to get the verbal slots and adjective slots. The slot label corresponding to the tokens marked as *VB*, *VBP*, *VBZ*, *VBN* and *VBG* is regarded as "*ACT*". The slot label corresponding to the tokens marked as *JJ*, *JJR*, *JJS* is regarded as the "*STATE*". (3) By manually integrating and modifying the results of NER and Pos Tagging, 15 types of slot labels and 17 types of slot labels are defined for SAMSum dataset and MediaSum dataset. (4) Finally, we fine-tune the pre-trained slot filling model (Chen et al., 2019) to get the complete semantic slot information.

We follow Gliwa et al. (2019) to adopt 14,732/818/819 for training/validation/test split on SAMSum dataset. We employ the split, i.e., 443,596/10,000/10,000, for MediaSAM dataset following Zhu et al. (2021b). Other statistics of the two datasets are shown in Table 1.

4.2 Baselines

The following models are adopted as baselines: (1) Pointer-Generator model (PG) (See et al., 2017); (2) Transformer (Vaswani et al., 2017) (TRAN); (3) Topic-word Guided Dialogue Graph Attention model (TGDGA) (Zhao et al., 2020); (4) Dialogue Heterogeneous Graph Network (D-HGN) (Feng et al., 2020); (5) BART (Lewis et al., 2019); (6) BART-based Multi-View Seq2Seq model (M-BART) (Chen and Yang, 2020); (7) Structure-aware sequence-to-sequence model (S-BART) (Chen and Yang, 2021).

.

4.3 Evaluation Metrics

We use three automatic evaluation metrics to evaluate our models. The first is **ROUGE** scores (Lin,

Model	SAMSum Dataset					MediaSum Dataset				
	R-1	R-2	R-L	QGQA	SIC	R-1	R-2	R-L	QGQA	SIC
PG (See et al., 2017)	40.09	15.28	36.63	30.67	27.61	28.77	12.24	24.18	12.53	19.42
TRAN (Vaswani et al., 2017)	37.27	10.76	32.73	30.11	23.74	25.62	9.27	21.13	11.84	15.88
TGDGA (Zhao et al., 2020)	43.11	19.15	40.49	33.80	31.17	31.44	15.74	26.81	15.03	22.32
D-HGN (Feng et al., 2020)	42.03	18.07	39.56	31.28	29.43	30.03	14.52	26.09	13.92	21.14
BART (Lewis et al., 2019)	48.22	24.53	46.58	35.04	33.16	35.09	18.05	31.44	17.71	26.76
M-BART (Chen and Yang, 2020)	49.35	25.61	47.73	35.59	33.91	35.81	19.24	32.32	17.87	27.03
S-BART (Chen and Yang, 2021)	48.70	24.88	47.24	35.82	34.74	35.19	18.43	31.58	18.32	27.38
SSAnet	51.28	27.15	49.37	38.91	42.54	37.43	20.67	34.05	24.73	34.56

Table 2: Results in terms of ROUGE scores, QGQA, and SIC on test set of SAMSum and MediaSum datasets.

2004), the standard summarization quality metrics, which compare the word-level unigram, bigram, and longest common sequence overlap with the gold summary. Since the ROUGE scores have been criticized for their poor correlation with factual consistency, we use the QA-based model, i.e., **QGQA** (Wang et al., 2020), which have a high correlation with human judgements on factuality.

Except for factual consistency, factual completeness is also crucial to evaluate the factual correctness. To fill in this gap, we propose a new evaluation metric: slot Information Completeness (**SIC**). Formally, SIC is a recall of semantic slot information between a candidate summary and a gold summary, which is defined as follows:

$$\text{SIC} = \frac{\sum_{s \in S} Count_{match}(s)}{|S|} \quad (5)$$

where S stands for a set of slot values in the gold summary, $Count_{match}(s)$ is the number of values co-occurring in the candidate summary and gold summary, and $|S|$ is the number of values in set.

5 Results and Analysis

5.1 Main Results

Results on SAMSum As reported in Table 2, all baselines and our model are evaluated automatically with ROUGE scores, QGQA, and SIC. We can observe that, compared to simple sequence-to-sequence models (PG and TRAN), incorporating the extra information such as commonsense knowledge (D-HGN) and topic word information increases all scores. However, the performance of factual correctness metrics (QGQA and SIC) are very poor. Besides, although the utilize of pre-trained models, i.e., BART, M-BART, and S-BART, achieves a high level of ROUGE scores, the QGQA and SIC do not improve significantly, especially SIC. It suggests that the previous models only focus on improving the ROUGE scores, but ignore the exploring on factual consistency and factual completeness, which would cause the generated summaries with high ROUGE scores, but are incorrect and low-quality. It is worth noting that our SSAnet significantly boosts factual consistency measure and factual completeness measure (QGQA and SIC) by large margins, with improvements on ROUGE scores at the same time. This shows our model has the ability to improve the correctness of system-generated summaries via semantic slot information without sacrificing the informativeness.

Results on MediaSum As shown in Table 2, we notice that all results for the abstractive summarization models are especially lower than those on SAMSum dataset, because of the increase in the number of speakers and turns, and the high requirement of compression ratio. However, it is encouraging that the SSAnet surpasses the best performing model S-BART by 6.41 points and 7.18 points for QGQA and SIC scores, which shows that the semantic slot information guides the model to generate salient elements and plays an important role in reducing factual errors.

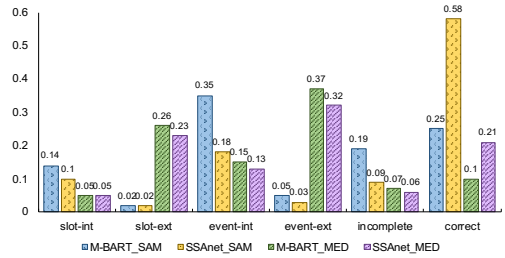

Figure 4: Fractions of examples in two datasets exhibiting different error types generated by M-BART and SSAnet.

5.2 Analysis of Error Types

To examine the performance of models, we qualitatively analyze the actual factual errors produced by them. We identify the factual errors through manual inspection and define two broad categories of errors: factual inconsistency and factual incompleteness. The errors of factual inconsistency occur at slot-level and event-level, each of which is further divided into intrinsic and extrinsic.

1. Factual Inconsistency

(1).**Slot-Int**:The tokens for slot values are in-

correctly replaced by other slot values also appeared in the original dialogue within the same type of slot label, i.e., "at (TIME) 6 (NUMBER) pm (TIME)"→"at (TIME) 8 (NUMBER) pm (TIME)".

(2).**Slot-Ext**:The tokens for slot values do not present in the original dialogue, i.e., hallucination.

(3).**Event-Int**:Due to the misinterpreting and wrongly integration salient elements, the semantic of the summary is in contradiction to the original dialogue. For example, "Sara baked cookies and Sally ate some"→"Sally baked cookies and ate".

(4).**Event-Ext**:The pragmatic meanings described in the summary are not mentioned in the original dialogue, such as "Bob buys an apple"→"Bob plays the basketball".

2. Factual Incompleteness

The salient elements (slot values) presented in the original dialogue are lost in the summary, such as "Mary is going to a bar on Green Street for the birthday party at 10 p.m"→"Mary will go to a bar for the birthday party".

We use the above taxonomy to annotate examples from SAMSum and MediaSum. For each dataset, we use the state-of-the-art model M-BART (Chen and Yang, 2020) to generate summaries followed by manual annotation (100 examples). Additionally, our model SSAnet is also annotated for error analysis in the same way.

Fig.4 shows the distribution of factual errors for these different settings. We first analyze the performance conducted by the M-BART on two dialogue summarization datasets. For SAMSam, we can see that 75% of the generated summaries contain factual errors. Of these 75%, the bulk of the produced errors is intrinsic, which is because that this dataset contains human-written gold summaries and is generally more reliable. Besides, the errors of factual inconsistent (35%) and factual incomplete (19%) are primarily event-related caused by sentence compression or fusion. For MediaSum, more summaries (90%) generated by M-BART model are factually incorrect and most of them (63%) is extrinsic. One reason for this is that the MediaSum data is automatically constructed according to topic descriptions and does not contain fact-related overviews. We then observe the results on two datasets trained by our SSAnet, which shows that most error types are reduced. Especially, the Event-Int error of factual consistency and the errors of factual incompleteness drop to 18% and 9% for SAMSum, and the Slot-Ext and Event-Ext errors related to factual consistency decreased by 3 and 5 points for MediaSum. It demonstrates that our methods effectively alleviate many kinds of factual errors in dialogue summarization.

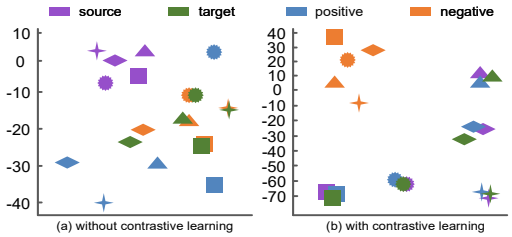

Figure 5: Visualization for the embedding space (a) without contrastive learning, and (b) with contrastive learning.

Model	R-1	R-2	R-L	QGQA	SIC
SSAnet	51.28	27.15	49.37	38.91	42.54
w/o SCA	49.53	25.81	48.04	37.28	40.71
w/o SBS	50.44	26.72	48.92	36.18	37.86
w/o SCA&SBS	48.86	25.03	47.11	35.03	36.33
w/o pos	51.10	26.86	49.52	38.69	42.33
w/o neg	50.88	26.71	49.20	38.63	42.38
w/o pos&neg	49.82	26.49	48.64	38.57	42.16

Table 3: Ablation studies for slot-level mask cross-attention (SCA), slot-driven beam search (SBS), and contrastive learning on test set of SAMSum dataset.

5.3 Ablation Study

As shown in Table 3, we first explore the contributions of the slot-level mask cross-attention module and the slot-driven beam search algorithm on SAMSum dataset. We can see that removing any components leads to the decline of performances. The removal of SCA almost has the same effect on R-1, R-2, R-L, QGQA, and SIC, which indicates that the SCA can comprehensively improve the n-gram overlap, factual consistency and factual completeness. However, deleting the SBS, that is, using the traditional beam search algorithm as the decoding strategy, makes little impact on ROUGE scores, but results in the decreases of 2.63% and 4.68% for QGQA and SIC. The huge impact on factual correctness evaluation metrics shows that our SBS can effectively reduce the factual errors in the generated summaries by controlling the decoding process. When the SCA and SBS are removed at the same time, the structure of the model is similar to BART and the performance of all metrics are also similar.

We then examine the contrastive learning framework. We can see that the adversarial perturbations, i.e., positive and negative pairs, can improve the performance to some extent. The visualization of this process is shown in Fig.5. Concretely, we apply the average pooling to the embeddings of the encoder outputs corresponding to source dialogue se-

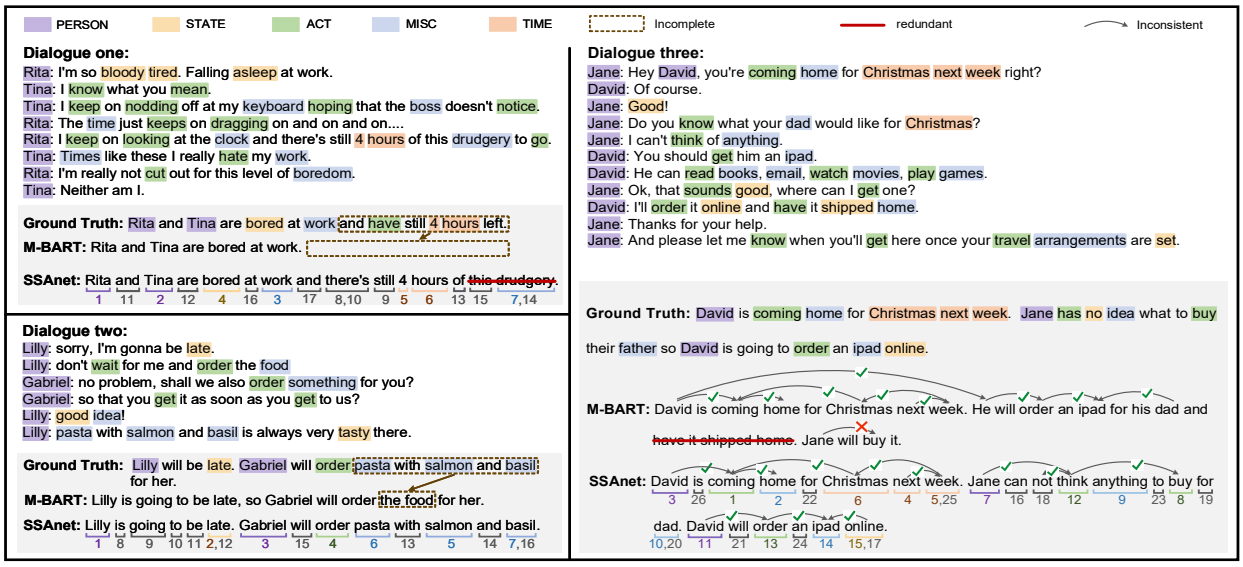

Figure 6: Sample summaries for dialogues from SAMSum dataset. The numbers underlined indicate the order in which the summary tokens are generated. "there's" stands for "there is". It maps to two tokens according to Byte Pair Encoding (BPE). Each sentence has an ending period, so the last word also maps to two tokens.

quences $\mathbf{H_s}$, the decoder outputs corresponding to target sequences $\mathbf{H_t}$, the additional positive examples $\tilde{\mathbf{H}}$, and the negative examples $\hat{\mathbf{H}}$. All of them are projected onto a two-dimensional space with t-SNE. As shown in Fig.5(b), the model pushes away the $\hat{\mathbf{H}}$ from the $\mathbf{H_t}$ and pulls the $\tilde{\mathbf{H}}$ to the embedding of the $\mathbf{H_s}$. However, for the model without contrastive learning, the $\mathbf{H_t}$ and $\tilde{\mathbf{H}}$ are far away from the $\mathbf{H_s}$, and the $\hat{\mathbf{H}}$ are very close to them as shown in Fig.5(a).

Model	Readability		Factualness	
	Flu.	Gra.	Con.	Com.
Ground Truth	4.82	4.79	4.30	4.05
M-BART	4.29	3.75	3.39	3.14
S-BART	4.20	3.68	3.57	3.22
SSAnet	4.11	3.64	3.98	3.82
w/o SCA	4.09	3.60	3.75	3.61
w/o SBS	4.32	3.79	3.61	3.39
w/o SCA&SBS	4.27	3.73	3.37	3.13

Table 4: Human evaluation on the Fluency (Flu.), Grammaticality (Gra.), Factual Consistency (Con.), and Factual Completeness (Com).

5.4 Human Evaluation

We run a human evaluation to investigate the quality of summaries. 100 samples are randomly selected from the test set of SAMSum and five annotators are hired from Amazon Mechanical Turk to rate the readability and factualness of ground truth, and summaries generated from M-BART, S-BART, and our models. Each annotator uses a Likert scale to score summaries from 1 (worst) to 5 (best) on readability—how fluent and grammatical the summaries are, and on factualness-whether the summaries are consistent with the original dialogue and the events described in the summary are complete.

As shown in Table 4, the generated summaries perform poor in grammaticality, factual consistency, and completeness. Compared with S-BART, the SSAnet and its variants have lower scores on readability, but have higher scores on factualness, which is due to the strategy of giving priority to generating salient elements by "filling-in-the-blanks" in the decoding process. Therefore, the fluency and grammaticality scores of SSAnet without SBS increase to 4.32 and 3.79. However, the SSAnet greatly improves the scores of factual consistency and completeness and is almost close to the performance of ground truths, which indicates that both of the SCA and SBS in our model play important roles in factual correctness. The outputs of three samples from SAMSum dataset can be found in Fig.6. Dialogue one and two show the ability of our model to solve factual incompleteness issues and Dialogue three mitigates the inconsistent facts in the generated summaries.

6 Conclusion

In this work, we propose a semantic slot guided adversarial sequence-to-sequence network for abstractive dialogue summarization, which utilizes the semantic slot information to improve the model architecture and decoding algorithm via the slot-level mask cross-attention mechanism and slot-driven beam search. A contrastive learning with adversarial perturbations is also introduced to assist the

training process. Experiments demonstrated the effectiveness of our proposed models in terms of both readability and factualness.

Acknowledgments

This work was partially supported by National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artificial Intelligence" Project No. MCM20190701, BUPT Excellent Ph.D. Students Foundation CX2021204.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W. Cottrell, and Julian McAuley. 2020. Rezero is all you need: Fast convergence at large depth.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6251–6258, Online. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*.

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

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

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling.

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.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-fact correction in abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9320–9331, Online. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

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

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.

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.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD19, page 166–175, New York, NY, USA. Association for Computing Machinery.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019.

Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. Association for Computational Linguistics.

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

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2020. Contrastive learning with adversarial perturbations for conditional text generation. *CoRR*, abs/2012.07280.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2018. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441, Santa Fe, New Mexico, USA. 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 Summarization 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 and Data Mining*, KDD '19, page 1957–1965, New York, NY, USA. Association for Computing Machinery.

Zhengyuan Liu, Angela Ng, Sheldon Lee, Ai Ti Aw, and Nancy 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.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

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

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

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.

Kaiqiang Song, Bingqing Wang, Zhe Feng, and Fei Liu. 2021. A new approach to overgenerating and scoring abstractive summaries.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. 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*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Lin Yuan and Zhou Yu. 2019. Abstractive dialog summarization with semantic scaffolds.

Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D. Manning, and Curtis Langlotz. 2020. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5108–5120, Online. Association for Computational Linguistics.

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

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021a. Enhancing factual consistency of abstractive summarization.

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

A Statistics of Slot Labels

ID	Relation Type	%
1	PERSON	7.10
2	TIME	2.50
3	NUMBER	1.10
4	LOCATION	0.40
5	TITLE	0.20
6	ORDINAL	0.10
7	ORGANIZATION	0.10
8	PRICE	0.06
9	PERCENT	0.05
10	URL	0.02
11	EMAIL	0.01
12	MISC	2.60
13	ACT	10.20
14	STATE	3.40
15	O	72.16

Table 5: Slot Label Types and Distribution in SAMSum dataset.

ID	Relation Type	%
1	PERSON	1.96
2	ORGANIZATION	1.65
3	TIME	1.48
4	NUMBER	0.56
5	EMAIL	0.47
6	WORK_OF_ART	0.28
7	FAX	0.19
8	PRICE	0.15
9	LOCATION	0.14
10	PERCENT	0.12
11	PRODUCT	0.05
12	LAW	0.02
13	LANGUAGE	0.01
14	MISC	0.06
15	ACT	10.97
16	STATE	5.05
17	O	76.84

Table 6: Slot Label Types and Distribution in MediaSum dataset.

B Training Details

Our methods are implemented with PyTorch (Paszke et al., 2019) and HuggingFace. We fine-tune the BART-large (Lewis et al., 2019) for all experiments. For parameters in the original BART encoder/decoder, we followed the default settings and set the learning rate 5e-5 with 120 warm-up steps. For graph encoder, we set the number of hidden dimensions as 1024, the number of layers as 2, and the dropout rate as 0.1. For the two extra cross-attention added to BART decoder layers, we set the number of attention heads as 4. The learning rate for parameters in newly added modules was 3e-4 with 60 warm-up steps. The model is fine-tuned for 20 epochs and the batch size is 128. At test time, the minimum lengths of generated summaries for two datasets are 35 and 20, and the beam size is 10.

C Human Evaluation Guidelines

In this subsection, we give details of the human evaluation guidelines.

C.1 Readability Annotation Guidelines

For readability, we make the annotators focus on how fluent and grammatical the summary is and we provide them the following guidelines:

1. First, the annotators judge whether the given sentence is complete or not. If the sentence is incomplete, the annotators will rate the scores as 1 both for fluency and grammaticality.

2. The annotators can understand the meaning of a complete sentence through their analysis, but there are many grammatical problems in the sentence. The annotators rate the scores as 2 or 3 both for fluency and grammaticality.

3. The annotator can easily understand the meaning of the sentence, and there are only minor grammatical problems in it. The annotators rate the scores as 4 or 5 both for fluency and grammaticality.

C.2 Factualness Annotation Guidelines

For factualness, we make the annotators focus on two types of unfaithful errors: (a) factual inconsistency, and (b) factual incompleteness. The guidelines are as follows:

1. We ask the annotators to check whether the given sentence is consistent with the source texts and whether the given sentence contains the complete fact descriptions.

2. If the matching degree is less than 30%, the annotators rate the scores as 1 or 2; if the matching degree is more than 30% and less than 60%, the annotators rate the scores as 3; if the matching degree is more than 60% and less than 100%, the annotators rate the scores as 4 or 5.