Topic-Aware Contrastive Learning for Abstractive Dialogue Summarization

Junpeng Liu [1]*, **Yanyan Zou**[2], **Hainan Zhang**[2], **Hongshen Chen**[2], **Zhuoye Ding**[2], **Caixia Yuan**[1] and **Xiaojie Wang**[1]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]JD.com, Beijing, China
{jeepliu, yuancx, xjwang}@bupt.edu.cn
{zouyanyan6,dingzhuoye}@jd.com
zhanghainan1990@163.com, ac@chenhongshen.com

Abstract

Unlike well-structured text, such as news reports and encyclopedia articles, dialogue content often comes from two or more interlocutors, exchanging information with each other. In such a scenario, the topic of a conversation can vary upon progression and the key information for a certain topic is often scattered across multiple utterances of different speakers, which poses challenges to abstractly summarize dialogues. To capture the various topic information of a conversation and outline salient facts for the captured topics, this work proposes two topic-aware contrastive learning objectives, namely coherence detection and sub-summary generation objectives, which are expected to implicitly model the topic change and handle information scattering challenges for the dialogue summarization task. The proposed contrastive objectives are framed as auxiliary tasks for the primary dialogue summarization task, united via an alternative parameter updating strategy. Extensive experiments on benchmark datasets demonstrate that the proposed simple method significantly outperforms strong baselines and achieves new state-of-the-art performance. The code and trained models are publicly available via https://github.com/Junpliu/ConDigSum.

1 Introduction

Online conversations have become an indispensable manner of communication in our daily work and life. In the era of information explosion, it is paramount to present the most salient facts of conversation content, rather than lengthy utterances, which is useful for online customer service (Liu et al., 2019a) and meeting summary (Zhao et al., 2019). This work focuses on *abstractive dialogue summarization*. To summarize dialogues, one simple way is to directly apply existing document summarization models to dialogues (Shang et al., 2018; Gliwa et al., 2019) or to employ hierarchical models to capture features from different turns of different speakers (Zhao et al., 2019; Zhu et al., 2020b). However, succinctly summarizing the dialogue is much more challenging.

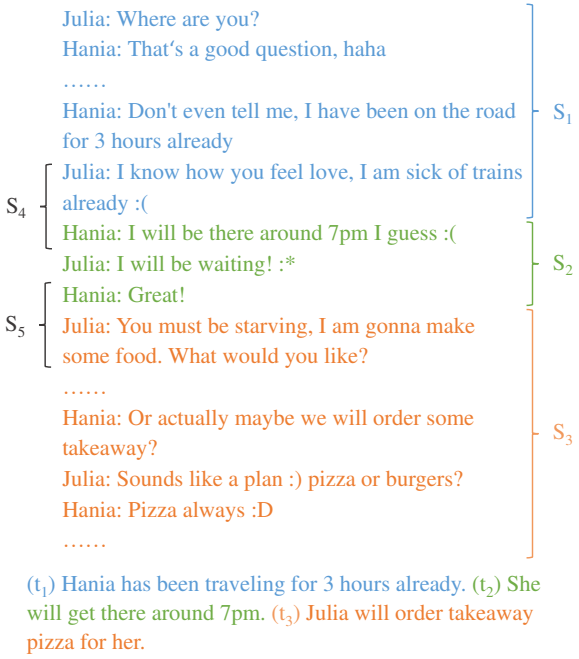

Figure 1: A dialogue and its paired summary. S_1, S_2, and S_3 stands for referred topic snippets, *current situation*, *time of arrival* and *food to eat*, respectively. The corresponding summary consists of three sentences t_1, t_2 and t_3. Each t_i corresponds to one snippet S_i ($i = 1, 2, 3$). S_4 and S_5 are inter-topic snippets.

The well-structured textual descriptions, such as news reports (See et al., 2017) and academic papers (Nikolov et al., 2018), often come from one single speaker or writer where the information flow is more natural and clearer with paragraphs and sections. Differently, consisting of multiple utterances from two or more interlocutors, the conversational content is in a complicated flow with information exchange and the focused topic can vary upon the conversation progression. On the other hand, the salient information for a specific topic is often scattered across multiple utterances

*Work partially done at JD.com.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1229–1243
November 7–11, 2021.

and can be presented separately. Exemplified by Figure 1, this dialogue touches three topics, *current situation, time of arrival* and *food to eat*, where the corresponding topic snippets are S_1, S_2 and S_3, respectively. The central ideas of each topic is summarized with one sentence, covering information from multiple utterances, i.e., t_1 for S_1, t_2 for S_2, and t_3 for S_3. We also observe that utterances residing in the same topic (e.g., S_1, S_2 and S_3) is inherently more coherent than those coming from different topics (e.g., the inter-topic snippet S_4 and S_5), which reveals the underlying relationships between topic and utterance coherence, also demonstrated by Glavaš and Somasundaran (2020).

Recent studies involves intrinsic information of dialogues to handle the challenges for summarizing dialogues, such as topic segment features (Liu et al., 2019b; Li et al., 2019; Chen and Yang, 2020), dialogue acts (Goo and Chen, 2018) and conversation stages (Chen and Yang, 2020). Although such existing models have demonstrated the effectiveness of the dialogue analysis on generating summaries, additional human efforts in data annotations or extra topic segmentation algorithms are necessary. For example, Goo and Chen (2018); Liu et al. (2019a) require extensive expert annotations on dialogue acts, while the knowledge of visual focus of each speaker and topic segment is a must for Li et al., 2019's work, which are both expensive and sometimes hard to obtain. Liu et al. (2019b); Chen and Yang (2020) need extra algorithms to obtain topic segment information, which works with the primary summarization model in a pipeline manner and thus may cause error propagation. Different from the structured text where a paragraph or a section can be treated as natural topic segment, it is difficult to accurately segment topics of dialogues.

Recall the inherent relationships between the topic and utterance coherence, this work proposes to implicitly capture the dialogue topic information by modeling the utterance coherence in a contrastive way. The coherence detection objective is constructed to push the model to focus more on snippets that are more coherent and likely contain salient information from the same topics. Further, since we aim to generate better summaries for each topic in a dialogue, we also introduce the *sub-summary generation objective*, which is expected to force the model to identify the most salient information and generate corresponding summaries. Note that both objectives are constructed in a contrastive way where no additional human annotations or extra algorithms are required. Such two contrastive objectives can be coupled with the primary dialogue summarization task via an alternating parameter updating strategy, resulting in our final model ConDigSum. Experiments on two dialogue summarization datasets demonstrate the effectiveness of our proposed contrastive learning objectives for dialogue summarization which achieves new state-of-the-art performances.

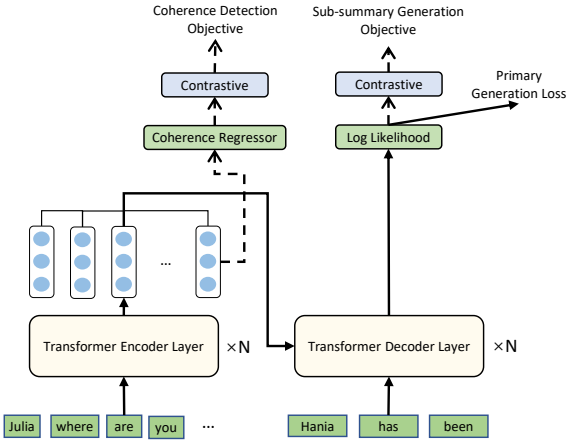

Figure 2: Model structure with contrastive objectives.

2 Proposed Method

2.1 Sequence-to-Sequence Learning

In this work, we frame the abstractive summarization task as a sequence-to-sequence learning problem. The sequence-to-sequence Transformer (Vaswani et al., 2017) is adopted as our backbone architecture, where the model takes as input the dialogue utterances and generates a corresponding summary. Specifically, given a dialogue $\mathcal{D} = (u_1, u_2, ..., u_{|\mathcal{D}|})$, consisting of $|\mathcal{D}|$ utterances, coupled with its corresponding summary $T_\mathcal{D} = (y_1, y_2, ..., y_{|T_\mathcal{D}|})$ in the length of $|T_\mathcal{D}|$, the goal is to learn the optimal model parameters θ and to minimize the negative log-likelihood:

$$\mathcal{L}^{\mathcal{D},T_\mathcal{D}} = \sum_{i=1}^{|T_\mathcal{D}|} -\log p(y_i|y_{1:i-1}, \mathcal{D}; \theta) \quad (1)$$

where $y_{1:i-1}$ denotes the first $i-1$ tokens of the output sequence (i.e., $y_{1:i-1} = (y_1, y_2, ..., y_{i-1})$). For a certain batch of dialogue-summary pairs $\mathcal{B} = (\langle \mathcal{D}_1, T_{\mathcal{D}_1} \rangle, \langle \mathcal{D}_2, T_{\mathcal{D}_2} \rangle, \ldots, \langle \mathcal{D}_{|\mathcal{B}|}, T_{\mathcal{D}_{|\mathcal{B}|}} \rangle)$, the negative log-likelihood is calculated as:

$$\mathcal{L}^{\mathcal{B}}_{main} = \frac{1}{|\mathcal{B}|} \sum_{\langle \mathcal{D}, T_\mathcal{D} \rangle \in \mathcal{B}} \mathcal{L}^{\mathcal{D},T_\mathcal{D}} \quad (2)$$

2.2 Contrastive Objectives

In this section, we introduce two contrastive objectives, coherence detection and sub-summary examination objectives, which can be considered as auxiliary tasks during training phase and reinforce the primary dialogue summarization task.

Coherence Detection Objective. The access to topic labels of dialogues often requires extra expert annotations or additional topic segment algorithms, which is expensive or may introduce error propagation. Considering the observation that text coherence is inherently related to the text topic (refer to Section 1), instead, we obtain the topical information of a dialogue by modeling the coherence change among utterances. The assumption behind this is that utterances within the same topic are more coherent than those spanning across different topics, based on which we construct the contrastive *coherence detection objective*.

To conduct contrastive learning, we construct positive-negative pairs with self-supervision. Recall that a dialogue consists of $|\mathcal{D}|$ utterances, i.e., $\mathcal{D} = (u_1, u_2, \ldots, u_{|\mathcal{D}|})$. We introduce a window comprising a subsequence of k ($k < |\mathcal{D}|$) utterances of a dialogue $\mathcal{D}$, as a *snippet*, denoted as $\mathcal{S}_k^{\mathcal{D}}$. For instance, $(u_j, u_{j+1}, \ldots, u_{j+k})$ is an example snippet for dialogue $\mathcal{D}$ where $j \in [1, |\mathcal{D}| - k]$ is an integer utterance index. Such a snippet is regarded as a positive example, while the corresponding negative snippet $\widetilde{\mathcal{S}_k^{\mathcal{D}}}$ is constructed by shuffling the order of sentences inside $\mathcal{S}_k^{\mathcal{D}}$. Given a pair of positive and negative examples, denoted as $\mathcal{P}_{co}^{\mathcal{D}} = (\mathcal{S}_k^{\mathcal{D}}, \widetilde{\mathcal{S}_k^{\mathcal{D}}})$, the contextual representations of each snippet can be obtained through the last layer of the Transformer encoder, denoted as $E_{\mathcal{S}_k^{\mathcal{D}}}$, $E_{\widetilde{\mathcal{S}_k^{\mathcal{D}}}}$, individually. Then we can calculate the coherence scores within a snippet by:

$$y_{\mathcal{S}_k^{\mathcal{D}}} = w_1 * E_{\mathcal{S}_k^{\mathcal{D}}} + b_1; \quad y_{\widetilde{\mathcal{S}_k^{\mathcal{D}}}} = w_1 * E_{\widetilde{\mathcal{S}_k^{\mathcal{D}}}} + b_1$$

where $w_1 \in \mathbb{R}^d$ and $b_1 \in \mathbb{R}$ are trainable parameters besides the original Transformer architecture, as depicted as *Coherence Regressor* in Figure 2. The normalization with a softmax layer is conducted to obtain the final coherence score:

$$[co(\mathcal{S}_k^{\mathcal{D}}), co(\widetilde{\mathcal{S}_k^{\mathcal{D}}})] = softmax([y_{\mathcal{S}_k^{\mathcal{D}}}, y_{\widetilde{\mathcal{S}_k^{\mathcal{D}}}}])$$

For a dialogue $\mathcal{D}$, there exist at least $|\mathcal{D} - k|$ contrastive snippet pairs, while, for simplicity, we randomly select $N_{co} < |\mathcal{D} - k|$ pairs for each epoch during training. The contrastive margin-based coherence loss is then calculated as:

$$\mathcal{L}_{co}^{\mathcal{D}} = \frac{1}{N_{co}} \sum_{n=1}^{N_{co}} \max(0, \delta_{co} - (co(S_{k,n}^{\mathcal{D}}) - co(\widetilde{S_{k,n}^{\mathcal{D}}})))$$

where δ_{co} is a margin coefficient by which we expect that the coherence score for the positive snippet is larger than the score for the negative one. k, N_{co} and δ_{co} are hyperparameters. For a certain batch of dialogue-summary pairs $\mathcal{B} = (\langle \mathcal{D}_1, T_{\mathcal{D}_1} \rangle, \langle \mathcal{D}_2, T_{\mathcal{D}_2} \rangle, \ldots, \langle \mathcal{D}_{|\mathcal{B}|}, T_{\mathcal{D}_{|\mathcal{B}|}} \rangle)$, the margin-based contrastive loss is calculated as:

$$\mathcal{L}_{co}^{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{\langle \mathcal{D}, T_{\mathcal{D}} \rangle \in \mathcal{B}} \mathcal{L}_{co}^{\mathcal{D}} \tag{3}$$

In this setting, we only use the dialogue while the summary is untouched. The coherence loss can be used to update the parameters in the encoder.

Algorithm 1 Snippet selection for a sub-summary

Input: A sub-summary $t_i \in T$, a dialogue $\mathcal{D}$ containing $|\mathcal{D}|$ utterances, sliding window size interval $[a, b]$
Output: $(S_{\text{pos}}^i, S_{\text{neg}}^i)$ for t_i

```
W = ∅
for w = a to b do
    for j = 1 to |D| − w do
        cand = D_{j,j+w}
        r(j, w) ← ROUGE(cand, t_i)
        W ← W ∪ cand
        j ← j + w/2
    w ← w + 1
j_best, w_best ← arg max_{j,w} r(j, w)
S^i_pos ← D_{j_best,(j_best+w_best)}
S^i_neg ← W \ S^i_pos
```

Sub-summary Generation Objective. The summary of a long dialogue always consists of multiple sentences each of which is regarded as a *sub-summary*. Considering the fact that one dialogue may contain more than one topics, we assume that each sub-summary is related to one topic. Hence, we introduce the contrastive *sub-summary generation objective*.

It is straightforward to obtain the sub-summaries by dividing the whole summary into single sentences via period symbols[1]. For simple illustra-

[1]More details are in the appendix.

tion, here we denote the corresponding target summary of a dialogue $\mathcal{D} = (u_1, u_2, \ldots, u_{|\mathcal{D}|})$ as $T_\mathcal{D} = (t_1, t_2, ..., t_m)$, where m is the number of sentences and each t_i is considered as a sub-summary. Given a sub-summary t_i, we can retrieve the most related snippet $\mathcal{S}^i_{\text{pos}}$ from the dialogue $\mathcal{D}$ according to the ROUGE-2 recall score (Lin, 2004). The detailed selection algorithm is presented in Algorithm 1. Given an integer window size $w \in [a, b]\,(0 < a \leq b < |\mathcal{D}|)$, we can slide the window over the dialogue $\mathcal{D}$ in the stride of half window size and obtain a set of candidate snippets $\mathcal{W}$. Enumerating each snippet candidate in $\mathcal{W}$ and calculating the ROUGE-2 recall score with the sub-summary t_i, we can get the optimal snippet scored the highest, which is selected as the most related snippet and regarded as the positive example S^i_{pos}. The corresponding negative example is randomly picked from the rest snippets in $\mathcal{W}$, denoted as S^i_{neg}. Now, we have constructed the contrastive sub-summary generation pairs $\{(S^i_{\text{pos}}, t_i), (S^i_{\text{neg}}, t_i)\}$. Like the primary dialogue summarization task, we also model the sub-summary generation objective as a sequence-to-sequence learning problem. Following Equation 1, the negative log-likelihoods are calculated as:

$$\mathcal{L}^{t_i}_{pos} = -\log(\prod_{j=1}^{|t_i|} p(t^i_j | t^i_{1:j-1}, \mathcal{S}^i_{\text{pos}}; \theta))$$

$$\mathcal{L}^{t_i}_{neg} = -\log(\prod_{j=1}^{|t_i|} p(t^i_j | t^i_{1:j-1}, \mathcal{S}^i_{\text{neg}}; \theta))$$

where t^i_j refers to the j_{th} token in t_i and $t^i_{1:j-1}$ stands for all preceding tokens before position j. The normalized scores after the softmax layer can be regarded as the irrelevance score to show how irrelevant a snippet is to a sub-summary:

$$[su(S^i_{\text{pos}}), su(S^i_{\text{neg}})] = softmax([\mathcal{L}^{t_i}_{pos}, \mathcal{L}^{t_i}_{neg}])$$

For a dialogue $\mathcal{D}$ paired with its summary $T_\mathcal{D}$, at least m contrastive pairs can be constructed, while, similar to the coherence case, we randomly select $N_{su} < m$ pairs for each epoch during training phase. Thus, we can construct a contrastive margin-based loss for dialogue $\mathcal{D}$:

$$\mathcal{L}^{\mathcal{D},T_\mathcal{D}}_{su} = \frac{1}{N_{su}} \sum_{n=1}^{N_{su}} \max(0, \delta_{su} - (su(S^n_{\text{neg}}) - su(S^n_{\text{pos}})))$$

Algorithm 2 Alternating Updating Strategy

Input: A batch of dialogue-summary instances $\mathcal{B}$

Coherence Task

$\mathcal{L}^\mathcal{B}_{co} = \frac{1}{|\mathcal{B}|} \sum_{\langle \mathcal{D}, T_\mathcal{D} \rangle \in \mathcal{B}} \mathcal{L}^\mathcal{D}_{co}$

$\theta \leftarrow \theta - \alpha w_{co} \frac{\partial L^\mathcal{B}_{co}}{\partial \theta}$

Sub-summary Task

$\mathcal{L}^\mathcal{B}_{su} = \frac{1}{|\mathcal{B}|} \sum_{\langle \mathcal{D}, T_\mathcal{D} \rangle \in \mathcal{B}} \mathcal{L}^{\mathcal{D},T_\mathcal{D}}_{su}$

$\theta \leftarrow \theta - \alpha w_{su} \frac{\partial \mathcal{L}^\mathcal{B}_{su}}{\partial \theta}$

Main Task

$\mathcal{L}^\mathcal{B}_{main} = -\frac{1}{|\mathcal{B}|} \sum_{\langle \mathcal{D}, T_\mathcal{D} \rangle \in \mathcal{B}} \mathcal{L}^{\mathcal{D},T_\mathcal{D}}$

$\theta \leftarrow \theta - \alpha w_{main} \frac{\partial \mathcal{L}^\mathcal{B}_{main}}{\partial \theta}$

where δ_{su} is a margin coefficient by which we would like the relevance score between a positive snippet and a sub-summary to be at least larger than the relevance score of the negative pair. a, b, N_{su} and δ_{su} are hyperparameters. For a certain batch of dialogue-summary pairs $\mathcal{B} = (\langle \mathcal{D}_1, T_{\mathcal{D}_1} \rangle, \langle \mathcal{D}_2, T_{\mathcal{D}_2} \rangle, \ldots, \langle \mathcal{D}_{|\mathcal{B}|}, T_{\mathcal{D}_{|\mathcal{B}|}} \rangle)$, the negative log-likelihood is calculated as:

$$\mathcal{L}^\mathcal{B}_{su} = \frac{1}{|\mathcal{B}|} \sum_{\langle \mathcal{D}, T_\mathcal{D} \rangle \in \mathcal{B}} \mathcal{L}^{\mathcal{D},T_\mathcal{D}}_{su} \quad (4)$$

The sub-summary objective can be used to update the parameters in the encoder and decoder.

2.3 Multi-Task Learning

The proposed two contrastive objectives can contribute to the primary dialogue summarization task during training phase, acting as auxiliary tasks. There are two options to combine the primary and auxiliary tasks: 1) summing the three objectives as a single one and update the model parameters using the summation loss; 2) alternatively update the model parameters using one of three objectives at each time. The empirical studies (Section 3.3) show that the alternating updating strategy performs better. Thus, in this work, we adopt the alternating parameter updating strategy, as shown in Algorithm 2. For a certain batch of dialogue-summary pairs, three objectives are adopted to update parameters in sequence. We first update the model parameters using the coherence objective, followed by the sub-summary and the primary generation objectives. The three objectives share the same learning rate α. Since the main focus is to generate better dialogue summaries with the help of auxiliary contrastive objectives, we give more attentions to the

primary task. Inspired by Dasgupta and Nambood-iri, 2016, to drive the auxiliary tasks to contribute to the primary one yet not to be dominate, we also introduce task-wise coefficients to each task, denoted as w_{co}, w_{su} and w_{main}, individually. Following experiments demonstrate the effectiveness of the alternating strategy and the introduced task-wise coefficients.

3 Experiment

3.1 Datasets

SAMSum contains natural message-like dialogues in English written by linguists, each of which is annotated with summary by language experts (Gliwa et al., 2019). There are 14,732 dialogue-summary pairs for training, 818 and 819 instances for validation and test, respectively.

MediaSum is a large-scale dataset for dialogue summarization, containing interview transcripts collected from National Public Radio (NPR) [2] and CNN [3], where the overview descriptions or discussion guidelines, coming with the transcripts, are considered as corresponding abstractive summaries (Zhu et al., 2021). The whole corpus contains 463.6K instances, with 10K each for validation and testing individually, and the rest is for training.

3.2 Implementation Details

As mentioned in Section 2.1, the sequence-to-sequence Transformer model is adopted as our backbone architecture, implemented using Fairseq toolkit[4] (Ott et al., 2019). To be specific, our model is initialized with a pre-trained sequence-to-sequence, i.e., BART (Lewis et al., 2020). Thus they share the same architectures, 6-layer encoder-decoder Transformer for $\text{BART}_{\text{BASE}}$ and 12-layer Transformer for $\text{BART}_{\text{LARGE}}$. Each layer in $\text{BART}_{\text{BASE}}$ has 16 attention heads, and the hidden size and feed-forward filter size is 1024 and 4096, respectively, resulting in 140M trainable parameters. Each layer in $\text{BART}_{\text{LARGE}}$ has 16 attention heads, and the hidden size and feed-forward filter size is 1024 and 4096, respectively, resulting in 400M trainable parameters. The dropout rates for all layers are set to 0.1. The optimizer is Adam (Kingma and Ba, 2015) with warmup. The learning rate α for SAMSum is 4e-5, 2e-5 for MediaSum. The maximum number of tokens for a certain batch is 800 and 1100 for SAMSum and MediaSum, individually. The margin coefficients δ_{co} and δ_{su} for the two contrastive objectives are always set to 1. Other hyper-parameters of our methods, including w_{co}, w_{su}, k, a, b are tuned on the validation set.

More implementation details and sensitivity tests for hyper-parameters are included in the appendix.

Model	R-1	R-2	R-L	BERTS
∗Lead3	31.4	8.7	29.4	-
∗PTGen	40.1	15.3	36.6	-
∗DynamicConv + GPT-2	41.8	16.4	37.6	-
∗FastAbs-RL	42.0	18.1	39.2	-
∗DynamicConv + News	45.4	20.7	41.5	-
Multiview BART	53.9	28.4	44.4	53.6
∗$\text{BART}_{\text{BASE}}$	46.1	22.3	36.4	44.8
∗BART	52.6	27.0	42.1	52.1
∗BART_{ORI}	52.6	27.2	42.7	52.3
$\text{CONDIGSUM}_{\text{BASE}}$	48.1	24.0	39.2	48.0
CONDIGSUM	**54.3**	**29.3**	**45.2**	**54.0**
w/o Sub-summary	53.8	28.3	44.1	53.5
w/o Coherence	53.9	28.6	44.2	53.5

Table 1: Results on SAMSum test split. ∗ indicates that the results are significantly different from ours ($p < 0.05$).

3.3 Evaluation

To evaluate our models, we utilized the ROUGE (Lin, 2004) to measure the quality of summary output generated by different models. We adopted the files2rouge[5] package based on the official `ROUGE-1.5.5.pl` perl script to get full-length ROUGE-1, ROUGE-2 and ROUGE-L F-measure scores. The recent popular automatic evaluation metric for text generation, BERTScore (Zhang et al., 2020b), is also presented for comparisons[6]. For simiplicity, we use R-1, R-2, R-L and BERTS to refer to ROUGE-1, ROUGE-2, ROUGE-L and BERTScore, respectively.

Baselines Lead3 is a commonly adopted method in the news summarization task, which simply takes the first three leading sentences of text as its summary. PTGen (See et al., 2017) extends sequence-to-sequence model with copy and coverage mechanisms. FastAbs-RL (Chen and Bansal,

[2] www.npr.org

[3] www.transcripts.cnn.com

[4] We empirically observed that different frameworks (e.g. Fairseq and Huggingface Transformer) may obtain different results under the same hyperparameter settings.

[5] https://github.com/pltrdy/files2rouge Note that the ROUGE scores might vary with different tookits.

[6] We use version 0.3.8, with default English setting (roberta-large_L17_no-idf_version=0.3.8(hug_trans=4.4.0)-rescaled).

Model	R-1	R-2	R-L	BERTS
∗Lead3	15.0	5.1	13.3	-
∗PTGen	28.8	12.2	24.2	-
∗UniLM	32.7	17.3	29.8	-
∗BART	34.7	17.7	30.9	30.7
∗BART_{ORI}	35.0	17.9	31.1	31.2
CONDIGSUM	**36.0**	**18.9**	**32.2**	**32.4**
w/o Sub-summary	35.5	18.7	31.9	32.0
w/o Coherence	35.5	18.6	31.7	31.9

Table 2: Results on MediaSum test split. ∗ indicates that the results are significantly different from ours ($p < 0.05$).

Mechanism	R-1	R-2	R-L	BERTS
Alternating updating	54.3	29.3	45.2	54.0
Summation objective	53.3	28.2	44.1	53.1

Table 3: Results of multi-task combination strategies.

Systems	1st	2nd	3rd	4th	MR
BART	0.14	0.12	0.31	0.43	3.03
Multiview BART	0.19	0.27	0.25	0.29	2.64
CONDIGSUM	0.26	0.32	0.23	0.19	2.35
Gold	0.41	0.29	0.21	0.09	1.98

Table 4: Human evaluation on SAMSum: proportions of rankings. MR: mean rank (the lower the better).

2018) first selects pivot sentences and then generates abstract summary with reinforcement learning. DynamicConv + GPT-2/News (Wu et al., 2019) proposes a lightweight dynamic convolutions to replace the self-attention modules in the Transformer layers. UniLM (Dong et al., 2019) is a unified language model which can be used for both natural language understanding and generation tasks. BART (Lewis et al., 2020) is a pre-trained encoder-decoder Transformer model, with two versions $\text{BART}_{\text{BASE}}$ and $\text{BART}_{\text{LARGE}}$. For simplicity, we use BART to denote $\text{BART}_{\text{LARGE}}$. Multiview BART (Chen and Yang, 2020) incorporates mutli-view features to summarize dialogues, including global, discrete, topic and stage information of dialogues. BART_{ORI} finetunes the $\text{BART}_{\text{LARGE}}$ with its original pre-training tasks (i.e., sentence shuffling and text infilling) (Lewis et al., 2020), acted as auxiliary tasks like this work.

Results on SAMSum. The results on SAMSum dataset are listed in Table 1. Results of Lead3, PTGen, DynamicConv + GPT-2/News, and FastAbsRL are taken from Gliwa et al., 2019. Others are based on our implementations (see the appendix). As we can see that, according to ROUGE script our model CONDIGSUM significantly outperforms previous state-of-the-art models in the first block ($p < 0.05$), indicated by ∗, with regard to both ROUGE and BERT scores, which demonstrates the effectiveness of the proposed contrastive objectives. Comparing $\text{BART}_{\text{LARGE}}$ against BART_{ORI}, it is interesting to observe that treating the original pre-training objectives as auxiliary tasks during fine-tuning also leads to performance gains. However, our proposed contrastive objectives are more effective.

We also conducted an ablation study on the SAMSum dataset. The ROUGE-2 score drops 0.7 points after removing the coherence detection objective, while the performance drops 1 point by ignoring the sub-summary generation objective. Such a phenomenon indicates both proposed contrastive objectives help generate better summaries, while the sub-summary generation objective contributes more to the primary task, compared to the coherence detection objective. One reason is that the sub-summary generation objective and the primary summary task are both sequence-to-sequence learning problems, yet the coherence detection objective only affects the encoder part.

Results on MediaSum. Table 2 shows the results on the MediaSum dataset. Results of PTGen and UniLM are reported by Zhu et al., 2021. Similar to SAMSum, CONDIGSUM also outperforms all the baseline models. The ablation study on the MediaSum dataset shows both auxiliary tasks contribute to the primary task and the results of them are similar.

Impact of Different Multi-Task Combination Strategies. Table 3 listed the performance on SAMSum dataset adopting either the alternating parameter updating or the summation objective strategy. Compared to the $\text{BART}_{\text{LARGE}}$ baseline in Table 1, both strategies result in performance gains, while the alternating parameter updating strategy is more helpful. Hence, this work adopts the alternating parameter updating strategy.

Human Evaluation. Since the automatic evaluation mainly focuses on the semantic matching between the generated output and the ground truth, while the generated summaries may be disfluent or ungrammatical, we thus also elicit feedback from human efforts. We compared our proposed model with the human references, as well as two base-

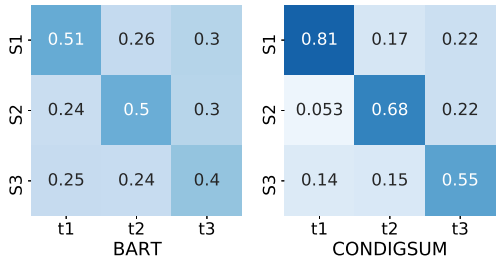

Figure 3: Visualization of how much a sub-summary is related to different snippets (the sum of every column is equal to 1). The result of CONDIGSUM is more concentrated on diagonal.

lines, BART (Lewis et al., 2020) and Multiview BART (Chen and Yang, 2020)[7]. 100 dialogues are randomly selected from the test split of SAMSum dataset. 10 participants are presented with a dialogue and its paired candidate summaries, including human references, generated outputs by three models. For each selected dialogue, they are asked to rank the candidate output from the best to worst with regard to *fluency* (is the summary fluent/grammatically correct?), *informativeness* (does the summary contains the most informative pieces of the dialogue?), and *succinctness* (does the summary express in an abstractive way?). Table 4 listed the proportions of different system rankings and mean rank (lower is better). The output of our CONDIGSUM is ranked as the most appropriate summary for 26% of all cases. Overall, we obtain lower mean rank than the other two systems but still lags behind the Gold one.

3.4 Case Study and analysis

How do coherence and sub-summary objectives work? Firstly, we compared the coherence scores predicted by our CONDIGSUM model of intra-topic snippets and inter-topic snippets. Taking the dialogue in Figure 1 from the test split of SAMSum dataset as an example, coherence scores of intra-topic snippets S_1, S_2 and S_3 are 1.37, 2.17 and 3.12, respectively, while the scores of inter-topic snippets S_4 and S_5 are much lower (-0.15 and -5.64, individually).[8] This indicates that the coherence detection objective does help the model capture the topical information of the dialogue. On the other hand, we tried to find out how the sub-summary generation objective affects the generation of summaries. For the same dialogue, we calculated

[7]Outputs are publicly available at https://github.com/GT-SALT/Multi-View-Seq2Seq

[8]Refer to the appendix for the illustration.

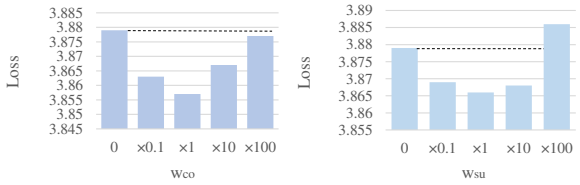

Figure 4: Impact of different values of task-coefficients of coherence detection (left) and sub-summary generation (right) objectives on the validation loss of the primary dialogue summarization task.

the sequence-to-sequence loss of snippet-summary pairs $\{(S_i, t_j), i, j \in \{1, 2, 3\}\}$ by feeding each snippet-summary pair into the trained model (the snippet for encoder and the summary for decoder). The log-likelihood loss was then transformed to represent the correlation score between a snippet and a sub-summary (a lower loss means a higher correlation). Figure 3 visualizes how much one sub-summary is related to different snippets (i.e., every column). The results of our CONDIGSUM model were more concentrated on the diagonal than those of BART, which proves that our sub-summary generation objective indeed forces the model to pay more attention to the most salient fact and generate more relevant summaries.

3.5 How does task-wise coefficients affect primary task?

In order to make it easier to observe how the task-wise coefficients affect the primary task, we only consider one contrastive objective at each time, by removing either the sub-summary generation objective or the coherence detection objective as well as scaling down and up the optimal values of the task-coefficients, w_{co} and w_{su}, based on the optimal values (denoted as $\times 1$). The values of the primary summrization loss on SAMSum dataset with different task-coefficients are depicted in Figure 4. We can observe that the primary loss increases with either larger or smaller task-coefficients. Assigning larger weights to the auxiliary tasks will encourage the model to prefer auxiliary tasks and ignore the primary task, where the primary task converges to the sub-optimal point. However, auxiliary tasks assigned by too small weight numbers will fail to assist the model to capture the dialogue topic information.

Is the coherence detection objective actually topical-related? To quickly investigate the relationship between coherence detection objective and discourse structures, we constructed a set of

70 contrastive examples. Each example is constructed as follows: For a snippet s_1, consisting of utterances from the same topic in a dialogue, we randomly select an utterance u in s_1 and replace it with another utterance v from other topics, where the dialogue act types for u and v are the same. Therefore, we get a new snippet s_2. The encoder of our model is used to get the coherence scores for s_1 and s_2, respectively. We found that the average coherence scores(-0.73) for the original snippets(s_1) are higher than the scores for their counterparts(s_2) with replacements(-1.02). Though the two examples have the same dialogue act types, the coherence scores are different. From this, we think the coherence detection objective does capture topic-related information. Moreover, from our understanding, a dialogue's topic and its discourse structure can be interlaced. The coherence score distribution of dialogue can reflect the topic change and also correlate to the discourse flow, while our work mainly focuses on the first point.

Relation between the quality of summary and complexity of dialogues. We further investigated the relation between the quality of generated summaries with regard to the number of sub-summaries residing in a dialogue summary. The test split of SAMSum dataset was divided into two sets: a) **One**: the dialogue summaries that only contain one sub-summary; b) **More**: the dialogue summaries consisting of more than one sub-summaries. For each set, we calculated the averaged ROUGE-2 score over all elements. We include CONDIGSUM, BART (Lewis et al., 2020) and Multiview BART (Chen and Yang, 2020) for comparison, as listed in Figure 5. Our model performs better than two baselines under both circumstances. In addition, under the **One** situation, CONDIGSUM outperforms Multiview BART by 0.41 ROUGE-2 point, yet the difference is expanded to 1.28 points under **More**. This increment indicates that our model significantly improves the quality of generated summaries when the dialogue summary comprises of more than one sub-summaries.

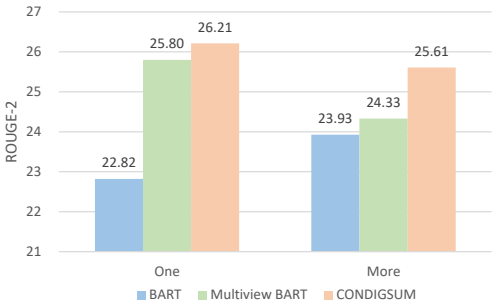

Figure 5: ROUGE-2 score of generated summaries for dialogues containing one or more sub-summaries.

4 Related Work

Document Summarization. Automatic document summarization aims to condense a well-structured document into its shorter form where the important information preserved. This task can be categorized into extractive and abstractive document summarization. The extractive summarizer learns to find the informative sentences from the input document as its summary, which can be viewed as a sentence problem (Kupiec et al., 1995; Conroy and O'leary, 2001). The features can be learned from LSTMs, CNNs or Transformers (Cheng and Lapata, 2016; Nallapati et al., 2017; Zhang et al., 2018, 2019; Liu and Lapata, 2019). The abstractive summarization task learns to generate summaries by rewriting the input document, which is a typical sequence-to-sequence learning problem. Sequence-to-sequence attentive LSTMs (Hochreiter and Schmidhuber, 1997; Bahdanau et al., 2015) and its extensions with copy mechanism (Gu et al., 2016), coverage mechanism (See et al., 2017) and reinforcement learning (Paulus et al., 2018) have shown effectiveness on summarizing the document. Recent studies have investigated the pretrained transformer models, like BERTAbs (Liu and Lapata, 2019), BART (Lewis et al., 2020), PEGASUS (Zhang et al., 2020a) and STEP (Zou et al., 2020).

The extractive and abstractive methods can be combined with reinforcement learning (Chen and Bansal, 2018), attention mechanisms (Gehrmann et al., 2018; Hsu et al., 2018) or in a pipeline manner (Pilault et al., 2020), while this work focuses on summarizing dialogue utterances from a sequence-to-sequence learning perspective.

Dialogue Summarization. The dialogue summarization task aims to summarize the dialogue content consisting of utterances from multiple speakers. Shang et al. (2018) proposed a simple multi-sentence compression technique to summarize meetings in an unsupervised fashion. Zhao et al. (2019); Zhu et al. (2020a) designed hierarchical model structures to capture features of conversational utterances from different turns.

The conversational analysis can also be unitized to generate the summaries for dialogue content. Liu et al. (2019b); Li et al. (2019) introduced the topical

information to the summarization process, while Liu and Chen (2019) took use of the key utterances and Goo and Chen (2018) leveraged the dialogue acts. Chen and Yang (2020) explicitly modeled conversational structures from four different views and then design a multi-view decoder to incorporate features from such four views to generate dialogue summaries. However, the additional information of conversational topics, key utterances, dialogue acts, and conversational structures requires human annotations, which is quite expensive or requires extra segment algorithms. Without requiring extra human effort or algorithms, this work proposes to introduce two contrastive learning objectives as auxiliary tasks during training.

Contrastive Learning. The application of contrastive learning for various tasks has been investigated recently, mainly in computer vision domain. The contrastive predictive coding (Oord et al., 2018) has been studied for data-efficient image recognition (Henaff, 2020). Without using specialized architectures or a memory bank, learning visual representations in a contrastive manner outperforms various baselines with self-supervised, semi-supervised and transfer learning (Chen et al., 2020). Khosla et al. (2020) proposed a fully-supervised contrastive loss which achieved new state-of-the-art results on the image classification task, surpassing the cross-entropy loss. This work also demonstrates that, compared to the traditional cross-entropy loss, the proposed supervised contrastive loss performs more stably to different hyperparameter settings, like data augmentations and optimizers. Moreover, Klein and Nabi (2020) introduced contrastive margin as regularizer for commonsense reasoning where a pairwise contrastive auxiliary prediction task is constructed. Fang et al. (2020) proposed to pre-train language models with contrastive self-supervised learning at the sentence level, which learns to predict whether two sentences originate the same one. Gunel et al. (2020) proposed a supervised contrastive learning objective which allows to work with cross-entropy and lead to significant performance gains. The contrastive learning is also introduced to learn the sentence embeddings (Gao et al., 2021). The above applications of contrastive learning are for computer vision or natural language understanding domains, while, in this work, we introduce the contrastive learning to the abstractive dialogue summarization task, which is a typical generation task.

5 Conclusion

Recent research progresses have present the effectiveness of dialogue studies (e.g., topical information and dialogue acts) on summarizing dialogues, while additional expert annotations or extra algorithms are required to obtain the knowledge. This work proposes a simple yet effective method, ConDigSum, that implicitly captures the topical knowledge residing in dialogue content by modeling the text coherence, yet no additional human annotations or segment algorithms are needed. We design two contrastive objectives as auxiliary task, i.e., coherence detection and sub-summary generation objectives,working together with the primary summarization task during training. An alternating parameter update strategy is employed to cooperate the primary and auxiliary tasks. Experiments on two benchmark datasets demonstrate the efficacy of the proposed model. Future directions include learning structured representations of information flow residing in dialogues and leveraging knowledge graphs to generate better dialogue summaries.

6 Ethical Considerations

Our simple yet effective abstractive dialogue summarization system could be used where there exists dialogue systems (two or multi-party dialogues). For example, it could be used for grasping the key points quickly or recapping on the salient information of online office meeting. In addition, the system can also be used for customer service, requiring employees to summarize the conversation records of customers' inquiries, complaints and suggestions.

The daily dialogue and media interview datasets used in this work are publicly available, and only for research purpose. There may exist biased views in them, and the content of them should be viewed with discretion.

Acknowledgements

Yanyan Zou and Xiaojie Wang are the corresponding authors. We would like to thank anonymous reviewers for their suggestions and comments. The work was supported by the National Natural Science Foundation of China (NSFC62076032) and the Cooperation Project with Beijing SanKuai Technology Co., Ltd. We would also like to thank all annotators who have contributed to the case study, especially Rizhongtian Lu.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.

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*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR.

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*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

John M Conroy and Dianne P O'leary. 2001. Text summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*.

Riddhiman Dasgupta and Anoop M Namboodiri. 2016. Leveraging multiple tasks to regularize fine-grained classification. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3476–3481. IEEE.

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

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Goran Glavaš and Swapna Somasundaran. 2020. Two-level transformer and auxiliary coherence modeling for improved text segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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*.

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

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*.

Olivier Henaff. 2020. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*. PMLR.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Tassilo Klein and Moin Nabi. 2020. Contrastive self-supervised learning for commonsense reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7517–7523.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation,

and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

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*.

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

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019a. Automatic dialogue summary generation for customer service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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)*.

Zhengyuan Liu and Nancy Chen. 2019. Reading turn by turn: Hierarchical attention architecture for spoken dialogue comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Zhengyuan Liu, A. Ng, Sheldon Lee Shao Guang, AiTi Aw, and Nancy F. Chen. 2019b. Topic-aware pointer-generator networks for summarizing spoken conversations. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 814–821.

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*.

Nikola I. Nikolov, Michael Pfeiffer, and Richard H. R. Hahnloser. 2018. Data-driven summarization of scientific articles. *CoRR*, abs/1804.08875.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

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

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

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*.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

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

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*.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Zhou Zhao, Haojie Pan, Changjie Fan, Yan Liu, Linlin Li, Min Yang, and Deng Cai. 2019. Abstractive meeting summarization via hierarchical adaptive segmental network learning. In *The World Wide Web Conference*, pages 3455–3461.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. MediaSum: A large-scale media interview dataset for dialogue summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, pages 5927–5934, Online. Association for Computational Linguistics.

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020a. End-to-end abstractive summarization for meetings. *arXiv e-prints*, pages arXiv–2004.

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

Yanyan Zou, Xingxing Zhang, Wei Lu, Furu Wei, and Ming Zhou. 2020. Pre-training for abstractive document summarization by reinstating source text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

A Experiment

A.1 Dataset

We also show detailed statistics about such two datasets, SAMSum (Gliwa et al., 2019) and MediaSum (Zhu et al., 2021), with regard to average tokens, utterances and speakers, as showed in Table 5. It is straightforward that the dialogue in MediaSum is much longer than the one in SAMSum, yet the corresponding summary is much shorter.

A.2 Implementation Details

The sequence-to-sequence Transformer model is adopted as our backbone architecture, implemented using Fairseq toolkit[9] (Ott et al., 2019). To be specific, our model is initialized with a pre-trained sequence-to-sequence, i.e., BART (Lewis et al., 2020). Thus they share the same architectures, 6-layer encoder-decoder Transformer for $\text{BART}_{\text{BASE}}$ and 12-layer Transformer for $\text{BART}_{\text{LARGE}}$. Each layer in $\text{BART}_{\text{BASE}}$ has 16 attention heads, and the hidden size and feed-forward filter size is 1024 and 4096, respectively, resulting in 140M trainable parameters. Each layer in $\text{BART}_{\text{LARGE}}$ has 16 attention heads, and the hidden size and feed-forward filter size is 1024 and 4096, respectively, resulting in 400M trainable parameters. The dropout rates for all layers are set to 0.1. The optimizer is Adam (Kingma and Ba, 2015) with warmup.

For SAMSum dataset, the learning rate α is 4e-5, and the maximum number of tokens in each batch is 800. The model is trained for 3 epochs. Each epoch takes around 0.7 hours on single Tesla P40 GPU. The window size k of the coherence detection objective is tuned over 5 to 15, with a stride of 2. The optimal value is 14. The lower bound of sliding window size for the sub-summary generation objective is selected from [1, 5], with the difference between lower and upper bounds set to 20. The optimal values for w_{co} and w_{su} are 0.005 and 0.0001 individually. The number of contrastive pairs for each sample, i.e., N_{co} and N_{su}, is equal to 2.

For MediaSum dataset, the learning rate α is 2e-5, and the maximum number of tokens in one batch is 1100. The model is trained for 4 epochs, each of which takes around 15 hours on four Tesla V100 GPUs. Similar to SAMSum, for the coherence detection objective, the window size k is 10, and the task-wise coefficient w_{co} is 0.00005. The sliding window size interval of the sub-summary generation objective is $[1, 5]$, with the task-wise coefficient w_{su} of 0.00005. For simplicity, the number of contrastive pairs for each sample, i.e., N_{co} and N_{su}, is equal to 1. Following Zhu et al. (2021), we add interlocutors information before concatenating utterances, and then truncate the dialogues to keep only first 1024 tokens as input. All experiments were conducted on either Tesla P40 GPUs (24GB) or Tesla V100 GPUs (16GB).

[9] We empirically observed that different frameworks (e.g. Fairseq and Huggingface Transformer) may obtain different results under the same hyperparameter settings.

A.3 Construction of Sub-summary

All sub-summaries are constructed from ground-truth summaries following a pre-processing procedure. We only consider dialogues whose ground-truth summary consists of at least two sentences and filtered the sentences in ground-truth summaries that have no good match with any snippets in original dialogues in terms of ROUGE score. We also tried to take BertScore as the selection metric of snippets, but ROUGE was finally adopted because there is barely any difference between them and the cost of BertScore was much larger.

A.4 Results

The output of MultiviewBART (Chen and Yang, 2020) is publicly available at https://github.com/GT-SALT/Multi-View-Seq2Seq. Since the ROUGE scores may vary due to different toolkits, to make fair comparisons with our model, we recalculated the ROUGE scores on the output of MultiviewBART using the files2rouge[10], same as ours.

A.5 Performance on the Validation Set

The performance on the validation split of SAMSum and MediaSum is listed in Table 6 and 7, respectively.

A.6 Sensitivity tests

To explore the effects of the hyper-parameters of our methods, we conducted sensitivity tests on validation split of SAMSum. Generally, there is an optimal value reaching at highest ROUGE scores, while too small or too large values hamper performance. $*$ indicates the best setting according to the validation set.

[10] https://github.com/pltrdy/files2rouge

Datasets	DialogToken	UtterToken	SummaryToken	DialogUtter	Speaker
SAMSum	93.8	9.5	20.3	9.9	2.2
MediaSum	1,553.7	51.7	14.4	30.0	9.2

Table 5: Data statistics of dialogue summarization datasets. *DialogToken*, *UtterToken* and *SummaryToken* stand for the average number of tokens in dialogues, utterances and summaries, respectively. *DialogUtter* is the average number of utterances in dialogues. The last column lists the average number of speakers in dialogues.

Model	R-1	R-2	R-L
$\text{BART}_{\text{BASE}}$	48.7	25.2	39.1
BART	54.0	28.8	44.0
BART_{ORI}	53.7	28.2	43.5
$\text{ConDigSum}_{\text{BASE}}$	50.7	26.9	41.6
ConDigSum	55.3	30.5	45.5
w/o Sub-summary	54.9	29.5	44.6
w/o Coherence	54.8	29.6	44.9

Table 6: Results on SAMSum validation split.

Model	R-1	R-2	R-L
BART	34.9	17.8	31.0
BART_{ORI}	35.0	17.8	31.0
ConDigSum	35.6	18.7	31.9
w/o Sub-summary	35.4	18.5	31.8
w/o Coherence	35.3	18.4	31.6

Table 7: Results on MediaSum validation split.

A.7 Case Study

A complete example showing coherence scores of different snippets and the generation loss of one sub-summary with respect to different snippets is shown in Figure 6.

k	R-1	R-2	R-L
2	54.9	29.8	45.0
4	54.8	29.6	44.8
6	54.7	29.3	45.0
8	54.8	29.5	44.9
10	54.7	29.5	44.8
14∗	55.3	30.5	45.5
18	54.6	29.6	45.0

Table 8: Sensitivity test of the coherence window k.

a	R-1	R-2	R-L
1	54.4	29.2	44.6
3	54.7	29.4	44.7
5∗	55.3	30.5	45.5
7	54.8	29.3	44.9

Table 9: Sensitivity test of the sub-summary window's lower bound a.

N_{co}	R-1	R-2	R-L
1	54.8	29.7	45.0
2∗	55.3	30.5	45.5
3	55.0	29.8	45.3

Table 10: Sensitivity test of the number of contrastive pairs for each sample N_{co}.

N_{su}	R-1	R-2	R-L
1	54.9	29.8	45.0
2∗	55.3	30.5	45.5
3	54.7	29.5	45.0

Table 11: Sensitivity test of the number of contrastive pairs for each sample N_{su}.

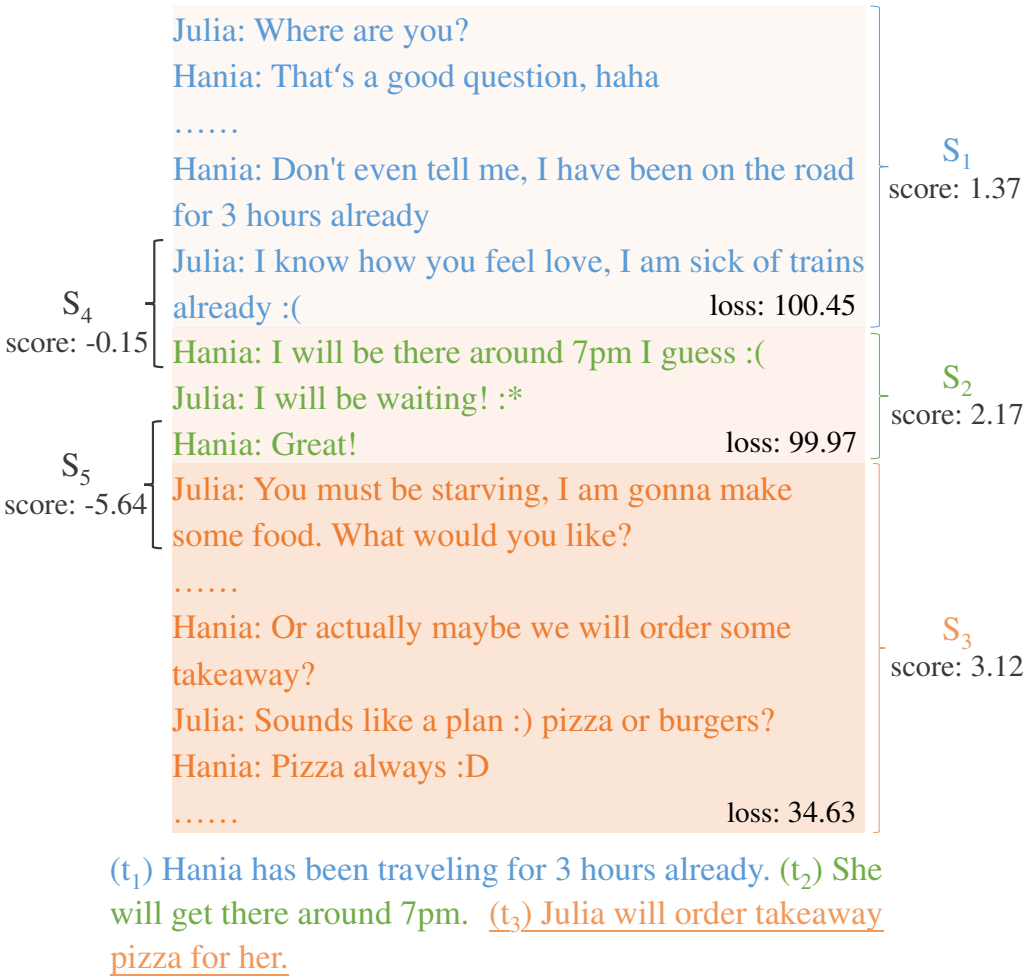

Figure 6: Coherence scores of snippets and visualization of how the sub-summary t_3 is related to different snippets S_i ($i \in \{1, 2, 3\}$). Darker background means a smaller loss and higher correlation between one snippet and the sub-summary t_3.