Salience-aware Dialogue Summarization via Parallel Original-Extracted Streams

Anonymous ACL submission

Abstract

In dialogue summarization, traditional approaches often concatenate utterances in a linear fashion, overlooking the dispersion of actions and intentions inherent in interactive conversations. This tendency frequently results in inaccurate summary generation. In response to this challenge, we formulate dialogue summarization as an extract-then-generate task. To tackle the extraction phase, we introduce an algorithm designed to identify **U**tterances **M**ost related to speakers' key **I**ntents (UMIs). These UMIs serve as labels to train an extraction model. Moving to the generation phase, we view a dialogue as parallel original-extracted streams. Correspondingly, we present a model named **R**ow-**C**olumn Fusion Dual-Encoders and **U**tterance **P**refix for Dialogue **S**ummarization, abbreviated as RCUPS[1], with the goal of enhancing the model's ability to discern utterances and align with our sentence-level extraction. RCUPS integrates the row-column wise fusion module, which amalgamates vector representations from a dual-branch encoder. In the decoding stage, an utterance-level prefix is strategically employed to emphasize crucial details, while weight decay is applied to non-UMIs to mitigate their influence. To assess the effectiveness of RCUPS, comprehensive experiments on SAMSum, DialogSum, and TODSum datasets show significant improvements over robust baselines.

1 Introduction

Conventional dialogue summarization methods treat the task as a sequence-to-sequence problem, which lack the ability to focus on crucial information in a dialogue, making models prone to inferring unfaithful summaries.

To address this challenge, we propose the extract-then-generate methodology. This approach mirrors human cognitive processes in dialogues, where key utterances (UMIs) are selected and summarized (Mao et al., 2022). Since dialogue summaries often center on "who did what" (Liu and Chen, 2021), extracting UMIs throughout the dialogue helps models discern the **K**ey **I**ntents (KIs) of speakers, improving summary accuracy. Previous research has explored methods combining extraction and summarization (Lebanoff et al., 2018; Xu and Durrett, 2019; Zhang et al., 2019a; Lebanoff et al., 2019; Zou et al., 2020; Bajaj et al., 2021; Zhang et al., 2021). These typically follow a sequential approach, as shown in Figure 2, generating summaries from extracted content. Other strate-

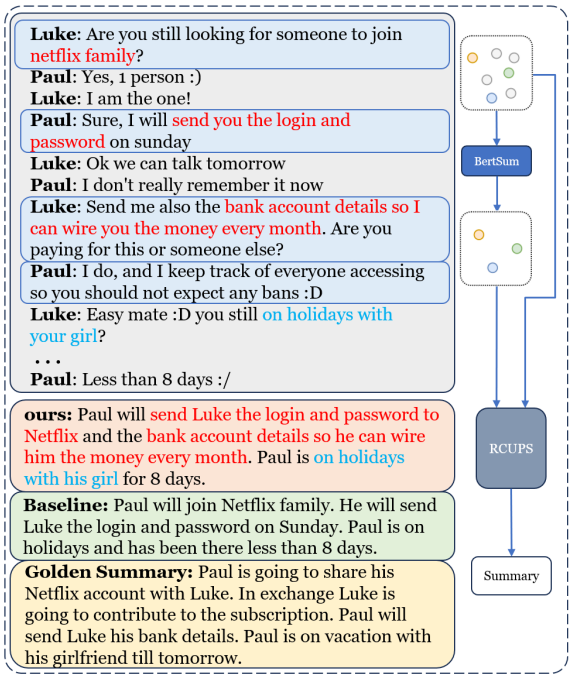

Figure 1: A dialogue summary samples generated by the baseline and the RCUPS model, reveal that the selected utterances effectively manifest the pertinent information in the summary, Meanwhile, RCUPS does not neglect the information in utterances that were not selected. In contrast, the baseline lacks emphasis on this particular information. Compared to the golden summary, our model produces superior outcomes than the baseline.

[1] https://anonymous.4open.science/r/Rcups-630B

gies, like entity chains (Narayan et al., 2021) or named entity sequences (Liu and Chen, 2021), do not focus specifically on capturing speakers' core intentions. In contrast, Yoo and Lee (2023) employ keyword extraction while retaining the original text, which may lead to contextually incoherent summaries due to discrete token combinations. These approaches generally concatenate extracted features with dialogue text, as depicted in Figure 2.

We propose an algorithm to select UMIs based on the summary, inspired by the Target Matching methodology (Zhang et al., 2022b). The algorithm operates on two assumptions: (1) long sentences in a dialogue contain rich and crucial information, and (2) sentences in the golden summary are semantically independent, following a "who did what" format, each representing a **K**ey **I**ntent (KI). This utterance-level matching enhances the accuracy and coherence of the dialogue representation. We use BertSUM (Liu, 2019) as a trainable extractive model, with specific training details provided in Section A.2.

The architectural framework of RCUPS is depicted in Figure 3. The dialogue text undergoes processing through three data streams: plain, utterance, and salient. Inspired by prior works (Humeau et al., 2019; Yang et al., 2022; Zhang et al., 2022a; Xie et al., 2022a), we employ a dual-encoder approach to encode these streams simultaneously. Integration of the row-column fusion module enhances information interaction between the streams, enabling the model to focus on dialogue KIs while retaining overall context awareness. During decoding, the model utilizes condensed information from the salient stream with the "extract-utterances" prefix. This directs attention to KIs. Subsequently, utterance weight is applied to reduce non-UMI scores, aiding in filtering redundant information for a more precise summary. Our main contributions can be summarized as follows:

- We present RCUPS, a model that combines two-dimensional fusion in encoding with information enhancement and weight decay in decoding. This enables the model to focus on key intents while retaining contextual information.
- We also introduce an efficient algorithm for extracting **U**tterances **M**ost related to speakers' Key **I**ntents (UMIs) from datasets lacking extractive annotations, using **K**ey **I**ntents (KIs) from the golden summary.
- We conducted comprehensive experiments on three datasets and discussed the pros and cons of large language models (LLM) in dialogue summarization.

2 Related Work

2.1 Dialogue Summarization

Dialogue summarization is a crucial research domain for extracting valuable insights from extensive conversations. The seminal SAMSum corpus by Gliwa et al. (2019), a high-quality, manually annotated dialogue dataset, has facilitated numerous baseline studies and advancements in this field. Researchers have adopted various graph-based strategies to model dialogue interactions, incorporating features like discourse graphs (Chen and Yang, 2021), heterogeneous graphs with commonsense knowledge (Xiachong et al., 2021), coreference graphs (Liu et al., 2021b), and static-dynamic graphs (Gao et al., 2023). Additionally, to capture dialogue nuances, methods such as named entities planning (Liu and Chen, 2021), speaker-aware self-attention (Lei et al., 2021), time-speaker streams (Xie et al., 2022b), and speaker-aware supervised contrastive learning (Geng et al., 2022) have been employed. To enrich dialogue understanding, Feng et al. (2021a) introduced an unsupervised DialoGPT annotator, and Chen et al. (2023) proposed using various levels of human feedback. Furthermore, Wang et al. (2023) presented a method for synthesizing query-based summarization triples, adding new dimensions to dialogue content exploration.

2.2 Extract-then-generate method

Recent studies employing the extract-then-generate method to produce more faithful summaries employ various extraction approaches. For instance, Lebanoff et al. (2018) utilizes Maximal Marginal Relevance (MMR) to select salient sentences, subsequently muting the attention score of corresponding sentences. On the other hand, Saito et al. (2020) train a saliency model to predict the saliency score of each sentence. Moreover, Zou et al. (2020) propose TDS, a foundational two-stage summarization model, comprising an utterance extractor and an abstractive refiner, which directly selects sentences based on their representations. Notably, these approaches typically sequentially connect the extractor's output to the decoder or generator, potentially

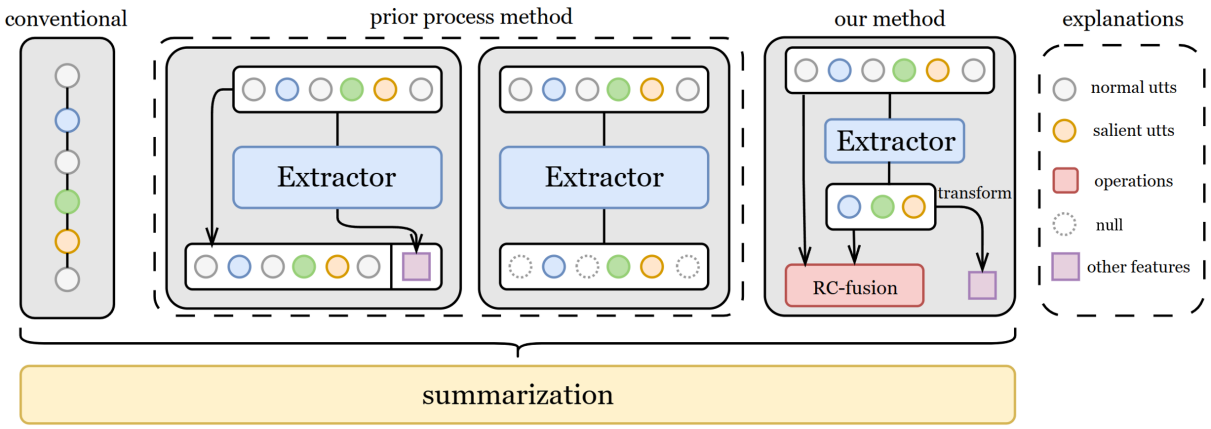

Figure 2: Traditional summarization approaches often resort to a straightforward concatenation of dialogues in chronological order. Meanwhile, prevailing methods in the field typically rely on either exclusively utilizing extracted sentences for generating content or extracting additional information, such as semantic features like keywords or entities. The subsequent step involves a mere concatenation of these extracted components with the dialogue context.In contrast, our method preserves the original text rather than discarding it. Furthermore, it transforms the UMIs into prefixes integrated into the decoding phase.

leading to the loss of contextual information from the original texts.

In contrast, RCUPS arranges the extractor's outcomes and original dialogue texts in parallel, thereby enabling the model to focus on the KIs conveyed by UMIs while retaining the original information.

Furthermore, beyond sentence extraction, prior research explores the utilization of other extracted features. For instance, Yoo and Lee (2023) perform keyword extraction using a BERT-based model and prepend the dialogue content with these words as prefixes for dialogue summarization. Another approach involves pre-training with entity chains composed of entity words as prompts to enhance abstract summarization capabilities (Narayan et al., 2021). Additionally, Liu and Chen (2021) enhance the controllability of the model's generation process and improve its ability to discern key named entities. Meanwhile, Ravaut et al. (2022) propose multiple summarization results as candidates, encoding dialogue content and candidates through the same encoder and concatenating these representations directly. In contrast, RCUPS adopts Row-column fusion to dynamically integrate original texts and UMIs.

3 Methodology

3.1 Problem Formulation

Given a dialogue $D^m = \{u_1, u_2, \cdots, u_m\}$ with m utterances, u_i denotes the i^{th} utterance in D^m, and its ground truth summary $S^n = \{s_1, s_2, \cdots, s_n\}$ with n sentences, s_j denotes the j^{th} sentence in summary S^n and $\hat{D^{m\prime}} = \{\hat{u_1}, \cdots, \hat{u_{m\prime}}\}$ denotes a selected subset (UMIs) of D^m and can be obtained with Algorithm 1. $m\prime$ represents the element number of the subset. Data sources D^m and $\hat{D^{m\prime}}$ are sent to a model to generate summaries. Our purpose is to maximize:

$$\max_{\theta} \sum_{i=1}^{|\Omega|} log_{p_\theta}(S_i^n | D_i^m, \hat{D_i^{m\prime}}) \quad (1)$$

where symbol θ represents the parameters of the model, and Ω refers to the training examples.

3.2 Extraction labels Generation

According to the content of the golden summary, a majority of the summaries comprise sentences in the format of "who did what," without explicit contextual connections. Inspired by the Target Matching approach (Zhang et al., 2022b), we similarly divide the summary into multiple sentence segments[2]. For each segment s_i, we calculate its ROUGE-1 score with the utterances in the corresponding dialogue and select the top k utterances based on this score, where k does not exceed a hyperparameter l. $\bigoplus$ represents the concatenation of utterances while maintaining their original order in the dialogue. Subsequently, we get the first k longest utterances in the dialogue. Finally, we take the union of the indices of these selected sentences. The process can be found in Algorithm 1.

[2] https://www.nltk.org/

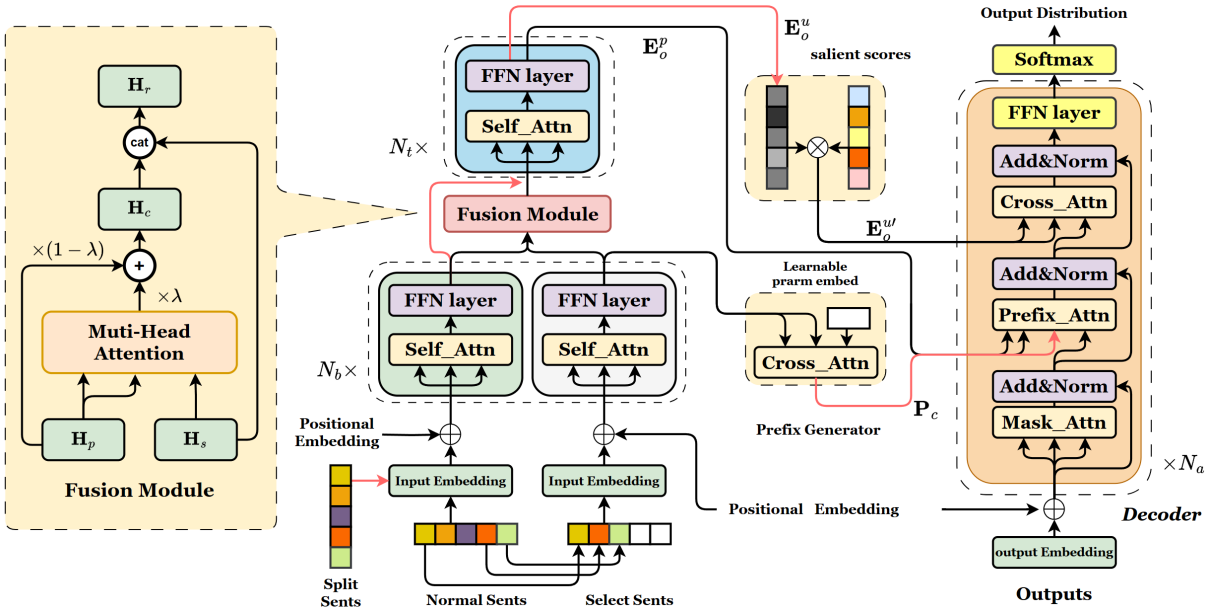

Figure 3: Overview of RCUPS, it processes dialogue text through three streams. The utterance stream is encoded sequentially by green and blue blocks. The other two streams pass through green and gray blocks, with their fused results forwarded to the blue block. A trade-off value of $\lambda = 0.7$ is found optimal. During training, all parameters are trainable.

In this paper, we employ BertSUM (Liu, 2019) without gram blocking to approximate the extractive labels. BertSUM is trained on the extraction labels from the training dataset and applied for inference on other datasets. The results obtained are then integrated into both the training and inference phases of RCUPS.

3.3 RCUPS Architecture

In this section, we introduce a model with **R**ow-**C**olumn Fusion Dual-Encoders and **U**tterance **P**refix for Dialogue **S**ummarizaion(RCUPS). RCUPS's backbone is based on BART (Lewis et al., 2019). An overview of RCUPS model is shown in Figure 3.

3.3.1 Original-Extracted Stream

To make our model capture the KIs in UMIs and reduce attention to redundant and distracting information, we introduce two additional input data streams. Consequently, the input can be summarized into the following three streams, with the Plain and Salient streams being part of the original input.

- **Plain stream**: This data stream treats the dialogue as a long sequence, which projects the dialogue onto the time dimension and we denote it as $\mathbf{H}_p$, preserving the order and temporal information of each token within the conversation.

- **Utterance stream**: Represent all the utterances as a vector. Here we use $\mathbf{E}_o^u$ to denote the set of all utterance vectors in a dialogue. This stream represents each individual utterance as a distinct vector. By segmenting the dialogue into separate utterances and converting each utterance into a vector, this stream allows for a more granular analysis of the dialogue, focusing on the properties and characteristics of each utterance independently.

- **Salient stream**: We use a pre-trained BERT model (Liu, 2019) to extract UMIs, and view all UMIs in a sequence, which we denote as $\mathbf{H}_s$. This stream aims to highlight and leverage the most significant pieces of information within the dialogue.

Through these data streams we hope to capture different levels of granularity and aspects of the dialogue. helping to provide a comprehensive representation of the dialogue from multiple perspectives.

The dual branch encoder (as shown in Figure 3) consists of two parts with a total layer number N_a, where $N_a = N_b + N_t$. Here, N_b represents the number of layers with two branches. Both branches contain an encoder module in BART which are de-

noted as $Branch_p(\cdot)$ and $Branch_s(\cdot)$ respectively, encoding the plain context and the UMIs, and we pad both stream to the same input length for the convenience of subsequent fusion operations and other processes. N_t represents the shared encoder layer number. This part is denoted as $Trunk(\cdot)$, aiming to better capture deep semantic information of fused vector representations.

$$\begin{aligned}
\mathbf{H}_p &= Branch_p([BOS], u_1, \cdots, u_m\}) \\
\mathbf{H}_s &= Branch_s([BOS], \hat{u_1}, \cdots, \hat{u_{m'}}\}) \\
u_i' &= \{[BOS], t_1^i, t_2^i, \cdots, t_{n_i}^i\} \\
\mathbf{H}_u &= Branch_p(\{u_1'^T, \cdots, u_m'^T\}^T) \\
\{\mathbf{H}_1^u, \cdots, \mathbf{H}_m^u\} &= Trunk(\mathbf{H}_u)
\end{aligned} \tag{2}$$

where t_j^i represents the j^{th} token in utterance u_i and n_i is the total token number of u_i. And $\mathbf{H}_i^u$ represents the set of all token vectors for the i^{th} utterance. We extract $\mathbf{H}_{i,0}^u$, which is the input special token [BOS], as the vector representation of the utterance. All $\mathbf{H}_{i,0}^u$'s are concatenated to a long vector sequence $\mathbf{E}_o^u = \{\mathbf{H}_{1,0}^u, \cdots, \mathbf{H}_{m,0}^u\}$.

3.3.2 Two Dimensional Fusion

The purpose of the Fusion Module (FM) is to fuse the outputs from $Branch_p(\cdot)$ and $Branch_s(\cdot)$. Hence, we propose a fusion module in both the row (r) and column (c) directions. The structure is shown in Figure 3.

FM first takes a cross-attention operation to give richer interactions of the two outputs (Humeau et al., 2019). Moreover, for preserving the original dialogue information $\mathbf{H}_p$ carries, FM does a weighted sum between the initial $\mathbf{H}_p$ and the output of cross-attention, where the weight coefficient (λ) is a hyperparameter. This process shown in equation 3 is called column-wise fusion. Here, we use $Attn(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ to indicate which information is used as query, key and value in the attention mechanism:

$$\mathbf{H}_c = (1-\lambda)\mathbf{H}_p + \lambda Attn(\mathbf{H}_s, \mathbf{H}_p, \mathbf{H}_p) \tag{3}$$

$$\mathbf{H}_r = [\mathbf{H}_c; \mathbf{H}_s] \tag{4}$$

Afterward, to better preserve the weights of the original UMIs, FM does a concatenation operation in another dimension as shown in equation 4, which is row-wise fusion. Then pass the output to a subsequent Encoder block ($Trunk(\cdot)$), which can be represented as follows:

$$\mathbf{E}_o^p = Trunk(\mathbf{H}_r) \tag{5}$$

3.3.3 UMIs Prefix Decoder (UPD)

Motivated by Ma et al. (2021); Liu et al. (2023), we improve the decoder of BART (Lewis et al., 2019) with a cross-attention projecting previously encoded vector sequence $\mathbf{H}_s$ into a short fixed-length prefix and an additional utterances-level cross-attention. UPD firstly initializes a learnable query embeddings $\mathbf{E} \in R^{Nd}$ and queries $\mathbf{H}_s$, projecting $\mathbf{E}$ to a fixed-length representation $\mathbf{P}_c$, where N is a hyperparameter and d is BART's token embedding dimension:

$$\mathbf{P}_c = Attn(\mathbf{E}, \mathbf{H}_s, \mathbf{H}_s) \tag{6}$$

Thus, these vectors can be viewed as the dense representation of $\mathbf{H}_s$, which carries the information of UMIs. Similar to Liu et al. (2023), $\mathbf{P}_c$ is projected into R^{LNd}, following which it is divided into L d-dimensional vector sequences, each having a length of N. These prefixes are aligned with the L layers within the transformer decoder. Subsequently, each of these is prepended to the transformer decoder's hidden state $\mathbf{H}_t$ in the corresponding layer, serving to iteratively emphasize the KIs, enhancing the UPD's focus on this informative segment. Specific operations can be referenced using the following formula:

$$\alpha_p = Attn(\mathbf{H}_t, [\mathbf{P}_c; \mathbf{E}_o^p], [\mathbf{P}_c; \mathbf{E}_o^p]) \tag{7}$$

In the second phase, we propose an importance label to forcefully modify the values of the utterances' vector representation, We use one-hot code to form a label of a dialogue, 1 for UMIs and 0 for others, where we denote w as the one-hot code label. Considering that non-UMIs carries contextual information, we don't completely zero the weights for the vectors associated with these utterances. Instead, we apply a softmax function to w which allocates a relatively small weight to these, reducing their impact during the decoding process.

$$\begin{aligned}
w\prime &= softmax(w) \\
\mathbf{E}_o^{u\prime} &= w\prime * \mathbf{E}_o^u \\
\alpha_u &= Attn(\mathbf{H}_t, \mathbf{E}_o^{u\prime}, \mathbf{E}_o^{u\prime})
\end{aligned} \tag{8}$$

where $\mathbf{E}_o^{u\prime}$ is the multiplication $\mathbf{E}_o^u$ and w', which is then fed into the second phase of the decoder.

Equation 8 illustrates the operation of this phase. UPD decodes the representation $E_o^{u\prime}$ that has undergone weight decaying. This stage acts as a denoising process, diminishing UPD's attention to redundant and distracting utterances.

4 Experiments

4.1 Baseline Models

BertAbs (Liu and Lapata, 2019) is an abstractive model with encoder initialized with BERT and trained with a transformer decoder. **BART** (Lewis et al., 2019) is an effective pre-trained model with a Transformer architecture for various tasks including summarization. **T5** (Raffel et al., 2020) is a versatile pre-trained model with a Transformer architecture for a wide range of tasks, including but not limited to summarization. **BART($\mathcal{D}_{\mathbf{ALL}}$)** (Feng et al., 2021b) uses the DialoGPT (Zhang et al., 2020) as an unsupervised dialogue annotator for keyword and topic information. **CONDIGSUM** (Liu et al., 2021a) proposes two topic-aware contrastive learning objectives to implicitly shift model topics and handle information scattering. **Coref-Attn** (Liu et al., 2021b) proposes to explicitly incorporate coreference information. **ATM** (Xie et al., 2022a) proposes a 2D view of dialogue based on a time-speaker perspective. **SICK++** (Kim et al., 2022) proposes to leverage the unique characteristics of dialogues sharing commonsense knowledge across participants to resolve the difficulties in summarization.

4.2 Evaluation Metrics and Datasets

For evaluation metrics, we follow existing dialogue summarization papers (Feng et al., 2021a) and use the ROUGE score (Lin, 2004) to assess summary quality, considering overlapping uni-grams, bi-grams, and the longest common subsequences. To avoid the limitations of automatic metrics alone (Stent et al., 2005), we also use embedding-based evaluations, including BERTScore (Zhang et al., 2019b) and BARTScore (Yuan et al., 2021), and conduct human evaluations. Dataset statistics are in Appendix A.1.

5 Results and Analysis

5.1 Automatic Evaluation

We compare our model with the baselines listed in Table 1. The proposed RCUPS achieves the best performances among other baselines on three datasets. Compared with BART$_{large}$, the original single-stream model, RCUPS improves the scores by 1.83, 1.38, and 1.81 for ROUGE-1, ROUGE-2, and ROUGE-L respectively on SAMSum. As for DialogSum, RCUPS boosts by 0.8, 0.11, and 9.13 for ROUGE-1, ROUGE-2, and ROUGE-L compared to BART$_{large}$. For TODSum, RCUPS brings improvements as well.

Method	R-1	R-2	R-L
SAMSum			
Oracle†	57.99	32.01	59.17
MV-BART	53.42	27.98	49.97
CONDIGSUM	54.30	29.30	45.20
Coref-Attn	53.93	28.58	50.39
SICK++	53.73	28.81	49.50
BART$_{large}$	52.96	28.62	54.38
RCUPS	**54.79**	**30.00**	**56.19**
DialogSum			
Oracle†	46.92	21.57	48.01
CODS	44.27	17.90	36.98
T5$_{large}$	45.22	18.96	37.72
SICK++	46.26	20.95	41.05
ATM	46.49	21.12	41.56
BART$_{large}$	45.95*	21.36*	38.72*
RCUPS	**46.75***	**21.47***	**47.85***
TODSum			
Oracle†	81.34	69.97	82.35
BertAbs	73.71	57.11	71.58
BART$_{large}$	73.96	60.66	72.02
RCUPS	**80.48**	**69.18**	**82.03**

Table 1: Automatic evaluation results. $*$ denotes the result using only the first reference in our evaluation. † denotes a greedy algorithm applied to select utterances whose combination maximizes the evaluation score against the gold summary, which is used as the upper bound of extractive methods.

Metrics	BART	RCUPS	ds$_b^{pt}$	ds$_u^{pt}$
BERTScore	91.67	**92.86**	90.59	91.03
BARTScore	-2.33	**-2.27**	-2.45	-2.39

Table 2: Semantic similarity evaluation on SAMSum. "ds" means DeepSeek. "pt" means prompt engineering. "b" means brief summaries. "u" means summarizing with UMIs

Since ROUGE is limited to assessing syntactical similarity at the token level, we also utilize BERTScore (Zhang et al., 2019b) and BARTScore (Yuan et al., 2021) to gauge the semantic con-

gruence between the generated summary and the ground truth on SAMSum. Results in Table 2 also confirm the superiority of RCUPS. Those results demonstrate the effectiveness of the additional modules that we proposed.

5.2 LLM Evaluation

5.2.1 Setup

For LLM evaluation, we use DeepSeek[3] (ds), a strong Mixture-of-Experts language model characterized by economical training and efficient inference, to generate summaries of the whole SAMSum test dataset. Our evaluation framework encompasses three dimensions: Faithfulness, Fluency, and Informativeness (Wang et al., 2023). Each dimension is assessed using a Likert Scale, ranging from 1 to 5. DeepSeek provides the scores for each dimension. The specific evaluation criteria can be found in Appendix A.5. Figure 4 illustrates the obtained results. We also compute the length of each summary generated by different methods, and the corresponding results are presented in Table 7. To ensure comprehensive evaluation, we employ the same criteria for human evaluation, details of which are provided in Appendix A.3. Furthermore, we conduct prompt engineering (Appendix A.4), incorporating our ideas into the evaluation process.

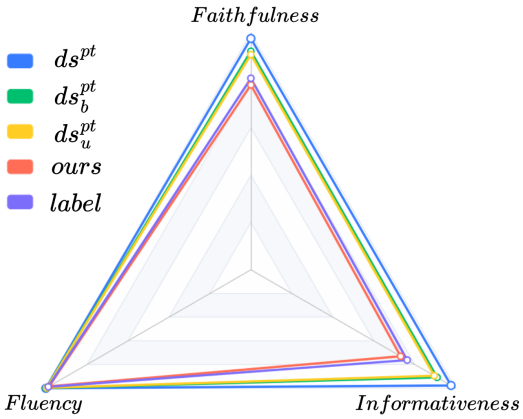

Figure 4: Scores of LLM evaluation on 3 aspects, In instructions. The specific data values in this figure can be referred to in Table 6

5.2.2 Pros and Cons Analysis

Regarding generated length, our model's output closely aligns with the label, with around 20 words. DeepSeek$^{pt}_b$ and DeepSeek$^{pt}_u$ exhibit similar generated lengths. DeepSeek^{pt} produces the longest summaries, with 50 words. This suggests that unless explicitly instructed otherwise, DeepSeek (or LLM) tends to generate more detailed summaries, as evident in the cases depicted in Appendix A.6.

The results in Figure 4 support this observation, revealing that DeepSeek^{pt} achieves near-perfect evaluation scores, particularly in terms of fluency, indicating the exceptional performance of large models in this regard. However, when evaluating metrics related to generation quality and accuracy, our model demonstrates a reasonable alignment with the labeled results, albeit with a noticeable discrepancy compared to DeepSeek^{pt}. It is worth noting that the other two DeepSeek sets also exhibit a decline in Faithfulness and Informativeness.

Hence, it can be inferred that in order to produce concise summaries, a model needs to make trade-offs during the generation process. This is why our model, which closely aligns with the human-written results, may have relatively lower scores in terms of informativeness and faithfulness compared to the LLM.

Method	Params	R-1	R-2	R-L
GPT-3-ft*	175B	53.4	29.8	45.9
ChatGPT^{pt}*	175B	32.7	12.3	24.7
ChatGPT$^{pt}_{rf}$*	175B	40.8	13.7	31.5
DeepSeek^{pt}	236B	36.4	13.2	39.2
DeepSeek$^{pt}_b$	236B	43.1	16.4	43.4
DeepSeek$^{pt}_u$	236B	46.3	18.5	45.7
RCUPS	570M	54.79	30.00	56.19

Table 3: The table shows ROUGE scores for ChatGPT and DeepSeek ($*$ from Wang et al. (2023)), plus our model's results. "rf" means reference summary lengths are given. "b" means brief summaries. "u" means summarizing with UMIs. "ft" means fine-tuning.

We compute ROUGE scores for the generated LLM results using various methods, as shown in Table 3. While ROUGE scores have limitations, they serve as a direct and objective evaluation metric. The results indicate that shorter summaries achieve higher scores compared to directly generated LLM outputs. DeepSeek with UMIs demonstrates a significant improvement in scores compared to other methods. Additionally, considering BartScore and BertScore (Table 2), our model's generated sentences exhibit greater similarity to the labels at the embedding level. In summary, LLMs' superior performance in faithfulness and informativeness

[3] https://github.com/deepseek-ai/DeepSeek-V2

stems from their tendency to summarize all conversation details comprehensively. This implies they do not prioritize brevity (indicating a need for more effective prompt engineering). While comprehensiveness is generally advantageous, it can be a drawback in dialogue summarization. A concise and effective summary naturally filters out unnecessary information. On the other hand, excessively lengthy summaries, which can be almost as long as the original conversation, fail to fulfill their primary purpose and render the summarization process ineffective.

5.3 Ablation Study

To investigate the effectiveness of each module, we make ablation studies on SAMSum from the perspectives of model input and structure.

Method	R-1	R-2	R-L
Input-wise			
Data stream			
-*w/o Salient stream*	54.03	28.67	54.58
-*w/o Utterance stream*	54.11	29.07	55.56
-RCUPS	**54.79**	**30.00**	**56.19**
Structure-wise			
Fusion module			
-*add*	53.97	28.62	55.38
-*w/o row wise*	51.74	25.73	52.65
-*w/o col wise*	54.13	29.15	55.59
-**value of** λ			
-$\lambda = 0.6$	54.49	29.35	55.63
-$\lambda = 0.7$	54.53	29.49	55.72
-$\lambda = 0.8$	54.51	29.75	55.88
Salient Score			
-*w/o label*	54.16	29.54	55.76
-*w/o softmax*	50.42	26.47	50.93
Salient utterance prefix			
-*w/o prefix*	54.53	29.49	55.72

Table 4: Ablations on SAMSum.

5.3.1 Input-wise Ablations

Effect of Using Two Additional Streams For RCUPS, the effect of feeding a single stream from either salient stream or utterance stream to the plain stream is inferior to the effect of feeding both streams to the plain stream simultaneously, as Table 4 shows, which indicates that the combination of the two streams brings additional improvements.

5.3.2 Structure-wise Ablations

Effect of Fusion Module We examine modifications to the fusion module, such as adding two streams and removing the row or column part, as shown in Table 4. Simple stream addition does not significantly enhance performance. Removing either part degrades performance, with the column part being more critical. This highlights the fusion method's role in model comprehension and generation. For λ values, ROUGE-1 scores rise then fall, suggesting a balanced fusion module optimally integrates streams.

Effect of Salient Scores Our experiments, summarized in Table 4, show that removing Salient Scores lowers ROUGE scores, and omitting the softmax function significantly degrades performance. This is because: (1) Non-UMIs, though not key, still provide contextual information for coherent summaries. (2) Zeroing out these vectors confuses the model and can collapse performance. Thus, the softmax operation is crucial for balancing salient and non-salient information, enhancing summary quality.

Effect of Salient Utterance Prefix The comparison presented in Table 4 highlights that the ROUGE score without the prefix module is lower than that of RCUPS. This observation underscores the significance of the prefix module in enriching the representations of salient information carried within the dialogue. By incorporating the prefix module, the model's attention to salient information during the decoding process is enhanced, leading to the generation of summaries that are more aligned with the factual content of the dialogue.

6 Conclusion

We introduce RCUPS, a method for dialogue summarization that utilizes an extractive approach for training labels and incorporates two crucial modules: row-column fusion and salient utterance prefix. The row-column fusion module enhances the encoding process by injecting salient information, while the salient utterance prefix module enriches decoding for generating concise summaries. RCUPS outperforms baseline models on three dialogue summarization datasets: SAMSum, DialogSum, and TODSum.

7 Limitations

Our work on RCUPS is subject to two main limitations that warrant consideration for future research endeavors.

The first limitation pertains to our initial approach in extracting UMIs using a TOP_k method. This method may inadvertently select redundant utterances, potentially impacting the quality of the generated summaries. Therefore, future efforts should focus on devising more effective extraction methods to improve the precision of UMIs selection.

Secondly, while the proposed extraction method enables RCUPS to demonstrate strong performance on three dialogue summarization datasets, we encounter constraints related to the maximum sequence length of BERT. As a result, for dialogue formats with extended lengths, such as meeting summarization, our current approach may encounter challenges in effectively extracting UMIs. Addressing this limitation could involve exploring alternative models or devising strategies to handle longer dialogue sequences more efficiently.

References

Ahsaas Bajaj, Pavitra Dangati, Kalpesh Krishna, Pradhiksha Ashok Kumar, Rheeya Uppaal, Bradford Windsor, Eliot Brenner, Dominic Dotterrer, Rajarshi Das, and Andrew McCallum. 2021. Long document summarization in a low resource setting using pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 71–80, Online. Association for Computational Linguistics.

Jiaao Chen, Mohan Dodda, and Diyi Yang. 2023. Human-in-the-loop abstractive dialogue summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9176–9190, Toronto, Canada. Association for Computational Linguistics.

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

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

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

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021b. Language model as an annotator: Exploring dialogpt for dialogue summarization. *arXiv preprint arXiv:2105.12544*.

Shen Gao, Xin Cheng, Mingzhe Li, Xiuying Chen, Jinpeng Li, Dongyan Zhao, and Rui Yan. 2023. Dialogue summarization with static-dynamic structure fusion graph. In *Annual Meeting of the Association for Computational Linguistics*.

Zhichao Geng, Ming Zhong, Zhangyue Yin, Xipeng Qiu, and Xuan-Jing Huang. 2022. Improving abstractive dialogue summarization with speaker-aware supervised contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6540–6546.

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.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Seungone Kim, Se June Joo, Hyungjoo Chae, Chaehyeong Kim, Seung-won Hwang, and Jinyoung Yeo. 2022. Mind the gap! injecting commonsense knowledge for abstractive dialogue summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6285–6300, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, W. Chang, and Fei Liu. 2019. Scoring sentence singletons and pairs for abstractive summarization. *ArXiv*, abs/1906.00077.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. *ArXiv*, abs/1808.06218.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *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*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Junpeng Liu, Yanyan Zou, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Caixia Yuan, and Xiaojie Wang. 2021a. Topic-aware contrastive learning for abstractive dialogue summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1229–1243, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shuai Liu, Hyundong Justin Cho, Marjorie Freedman, Xuezhe Ma, and Jonathan May. 2023. Recap: Retrieval-enhanced context-aware prefix encoder for personalized dialogue response generation. *ArXiv*, abs/2306.07206.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *ArXiv*, abs/1903.10318.

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)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Zhengyuan Liu and Nancy F. Chen. 2021. Controllable neural dialogue summarization with personal named entity planning. In *Conference on Empirical Methods in Natural Language Processing*.

Zhengyuan Liu, Ke Shi, and Nancy Chen. 2021b. Coreference-aware dialogue summarization. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 509–519, Singapore and Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. 2021. Luna: Linear unified nested attention. *Advances in Neural Information Processing Systems*, 34:2441–2453.

Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed Awadallah, and Dragomir Radev. 2022. DYLE: Dynamic latent extraction for abstractive long-input summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1687–1698, Dublin, Ireland. Association for Computational Linguistics.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan T. McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach 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. *CoRR*, abs/1912.01703.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Mathieu Ravaut, Shafiq R. Joty, and Nancy F. Chen. 2022. Towards summary candidates fusion. In *Conference on Empirical Methods in Natural Language Processing*.

Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, and Junji Tomita. 2020. Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models. *ArXiv*, abs/2003.13028.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *International conference on intelligent text processing and computational linguistics*, pages 341–351. Springer.

Bin Wang, Zhengyuan Liu, and Nancy F. Chen. 2023. Instructive dialogue summarization with query aggregations.

Feng Xiachong, Feng Xiaocheng, and Qin Bing. 2021. Incorporating commonsense knowledge into abstractive dialogue summarization via heterogeneous graph networks. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 964–975, Huhhot, China. Chinese Information Processing Society of China.

Keli Xie, Dongchen He, Jiaxin Zhuang, Siyuan Lu, and Zhongfeng Wang. 2022a. View dialogue in 2d: A two-stream model in time-speaker perspective for dialogue summarization and beyond. In *Proceedings of*

the 29th International Conference on Computational Linguistics, pages 6075–6088.

Keli Xie, Dongchen He, Jiaxin Zhuang, Siyuan Lu, and Zhongfeng Wang. 2022b. View dialogue in 2D: A two-stream model in time-speaker perspective for dialogue summarization and beyond. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6075–6088, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. In *Conference on Empirical Methods in Natural Language Processing*.

Yuanhang Yang, Shiyi Qi, Cuiyun Gao, Zenglin Xu, Yulan He, Qifan Wang, and Chuanyi Liu. 2022. Once is enough: A light-weight cross-attention for fast sentence pair modeling. *ArXiv*, abs/2210.05261.

Chongjae Yoo and Hwanhee Lee. 2023. Improving abstractive dialogue summarization using keyword extraction. *Applied Sciences*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Haoyu Zhang, Yeyun Gong, Yu Yan, Nan Duan, Jianjun Xu, Ji Wang, Ming Gong, and M. Zhou. 2019a. Pretraining-based natural language generation for text summarization. In *Conference on Computational Natural Language Learning*.

Kexun Zhang, Jiaao Chen, and Diyi Yang. 2022a. Focus on the action: Learning to highlight and summarize jointly for email to-do items summarization. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4095–4106, Dublin, Ireland. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Hassan Awadallah, Dragomir R. Radev, and Rui Zhang. 2022b. Summn: A multi-stage summarization framework for long input dialogues and documents: A multi-stage summarization framework for long input dialogues and documents. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Yusen Zhang, Ansong Ni, Tao Yu, Rui Zhang, Chenguang Zhu, Budhaditya Deb, Asli Celikyilmaz, Ahmed Hassan Awadallah, and Dragomir Radev. 2021. An exploratory study on long dialogue summarization: What works and what's next. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4426–4433, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lulu Zhao, Fujia Zheng, Keqing He, Weihao Zeng, Yuejie Lei, Huixing Jiang, Wei Wu, Weiran Xu, Jun Guo, and Fanyu Meng. 2021. Todsum: Task-oriented dialogue summarization with state tracking. *arXiv preprint arXiv:2110.12680*.

Yicheng Zou, Lujun Zhao, Yangyang Kang, Jun Lin, Minlong Peng, Zhuoren Jiang, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. 2020. Topic-oriented spoken dialogue summarization for customer service with saliency-aware topic modeling. *ArXiv*, abs/2012.07311.

A Appendix

A.1 Datasets

We evaluate our methods on three public dialogue summarization datasets: SAMSum (Gliwa et al., 2019), DialogSum (Chen et al., 2021), TODSum (Zhao et al., 2021). Detailed statistics are given in Table 5. Note that, in DialogSum, there are three reference summaries for each data sample, and we use only the first reference in our evaluation.

	SAMSum	DialogSum	TODSum
Train	14,732	12,460	7,892
Validation	818	500	999
Test	819	500	999
Avg.TD	9.9	9.49	14.1
Avg.SU	4.9	4.33	6.38

Table 5: Dataset Statistics for three benchmark datasets: SAMSum, DialogSum and TODSum. Avg.TD denotes the average turns of dialogue. Avg.SU denotes the average UMIs per dialogue

Algorithm 1 TOP_k utterance selecting

Input: T represents all sentences in a golden summary, U represents all utterances in a Dialogue. $T = \{t_1, t_2, \cdots, t_n\} U = \{u_1, u_2, \cdots, u_m\}$

Output: S

Let $S \leftarrow \Phi$.
$k \leftarrow LEN(U)/LEN(T)$.
if $k > l$ **then**
  $k = l$
end if
for $t_i \in T$ **do**
  $\tau \leftarrow ROUGE_1(t_1, T)$
  $\tau' \leftarrow Index(TOP_k(\tau, k))$
  $S \leftarrow S \bigoplus \tau'$
end for
$S' \leftarrow Index(l\ most\ long\ utterances\ in\ U)$
$S \leftarrow S \cup S'$
return S

A.2 Implementation Details

Our experiments are conducted using Pytorch (Paszke et al., 2019) on an NVIDIA RTX 3090 GPU with a 24GB memory. We initialize BART in our model with BART_{large} which has 16 attention heads, 1024 hidden size, and 12 Transformer layers for the decoder. For the encoder, the total layer number N_a is 12, and branch number N_b is 4. We set the batch size to 2 and the learning rate to 2e-5. The dropout rate is set to 0.1. We use AdamW optimizer (Loshchilov and Hutter, 2017) as our optimizing algorithm. During the test process, we employ beam search with size 5 to generate a more fluency summary. The training process took 8 hours, and the total number of parameters is 572M. All the parameters are trainable.

A.3 Human Evaluation

For human evaluation, we adopt three dimensions to assess the quality of each summary—Faithfulness, Fluency, and Informativeness (Wang et al., 2023). Each dimension is scored on a Likert Scale ranging from 1 to 5, with higher scores indicating superior performance. The specific evaluation criteria are displayed in section A.5 We utilized a total of 200 randomly selected samples from the test dataset of SAMSum for evaluation, with each sample accompanied by three summaries: baseline, golden summary (human-written), and our model-generated summary. Five volunteers participated in the evaluation process, yielding 198 responses. The mean scores for each metric were computed across all collected data, as presented in Table 6. To gauge the consistency of scoring among raters, we calculated Fleiss's Kappa scores, which ranged between 0.5 and 0.8. These scores indicate a moderate level of agreement between raters.

Models	**Fai.**	**Flu.**	**Inf.**
BART_{large}	4.28	4.46	4.11
labels_{he}	4.71	4.65	4.38
our_{he}	4.40	4.61	4.10
DeepSeek^{pt}	4.88	4.99	4.88
DeepSeek^{pt}_{b}	4.61	4.99	4.54
DeepSeek^{pt}_{u}	4.54	4.98	4.47
ours	3.90	4.92	3.65
labels	4.04	4.93	3.81

Table 6: human and LLM evaluation result. **Fai.** for Faithfulness. **Flu.** for Fluency. **Inf.** for Informativeness. "he" means human evaluation

Method	length
DeepSeek^{pt}	51.35
DeepSeek$^{pt}_{b}$	25.0
DeepSeek$^{pt}_{u}$	24.12
RCUPS	19.65
labels	20.02

Table 7: Average length of summaries generated by DeepSeek and our model, plus we append the average length of golden summaries(labels)

A.4 Generation Prompt

DeepSeek^{pt}:
Given the following dialogue, please summarize {Dialogue}.

DeepSeek$^{pt}_{b}$:
Your job is to summarize the given dialogue briefly. Only output the summary. Do not output the original dialogue or any other text.
Before you summarize, here is an example for you.
EXAMPLE:
Original Dialogue: {EXAMPLE}
Summary: {EXAMPLE_SUM}
Given the following dialogue, please summarize {Dialogue}.

DeepSeek$^{pt}_{u}$:
We introduce a concept called "sentences most relevant to the interlocutor's intent." These sentences are critical for summarization as they capture the essential points and intentions expressed by the speaker. You should use them as the basis for generating summaries. Only output the summary. Do not output the original dialogue, key sentences, or any other text.
Before you summarize, here is an example for you.
EXAMPLE:
Original Dialogue: {EXAMPLE}
Key Sentences: {KEY_EXAMPLE}
Summary: {EXAMPLE_SUM}.
Given the following dialogue, generate a summary based on these key sentences. please summarize {Dialogue}, and the key sentences you need to focus on: {Salient utterances}

A.5 Human and LLM Evaluation criteria

A.5.1 Faithfulness

- A score of 5 indicates that the content of the summary fully aligns with the information in the dialogue, with no errors.
- A score of 4 means that the majority of the content in the summary matches the dialogue, but there are minor discrepancies.
- A score of 3 suggests that some parts of the summary correspond to the dialogue, but there are some errors.
- A score of 2 indicates that most of the content in the summary does not align with the dialogue, with only a few points matching.
- A score of 1 means that the generated content is entirely incorrect.

A.5.2 Fluency

- A score of 5 signifies that the summary is very fluent, with no linguistic errors or misuse of words.
- A score of 4 indicates a few errors, but they are minimal.
- A score of 3 suggests there are some errors.
- A score of 2 means the sentences are somewhat incoherent.
- A score of 1 indicates significant difficulties and complete incoherence.

A.5.3 Informativeness

- A score of 5 indicates that the summary contains a wealth of information, very comprehensive.
- A score of 4 means there is a good amount of information with a few omissions.
- A score of 3 suggests that some important information is missing.
- A score of 2 indicates that the summary contains only a small amount of information.
- A score of 1 means that there is absolutely no information.

A.6 Cases

Hannah: Hey, do you have Betty's number?
Amanda: Lemme check
Hannah: <file_gif>
Amanda: Sorry, can't find it.
Amanda: Ask Larry
Amanda: He called her last time we were at the park together
Hannah: I don't know him well
Hannah: <file_gif>
Amanda: Don't be shy, he's very nice
Hannah: If you say so..
Hannah: I'd rather you texted him
Amanda: Just text him
Hannah: Urgh.. Alright
Hannah: Bye
Amanda: Bye bye

DeepSeek^{pt}: Hannah asked Amanda for Betty's phone number, but Amanda couldn't find it. Amanda suggested that Hannah ask Larry, who had called Betty before, and reassured Hannah that Larry is nice. Hannah expressed her preference for Amanda to text Larry instead. Amanda encouraged Hannah to text Larry herself.

DeepSeek$_b^{pt}$: Hannah couldn't find Betty's number and was advised by Amanda to ask Larry for it.

DeepSeek$_u^{pt}$: Hannah asked Amanda for Betty's number, but Amanda could not find it. Amanda suggested Hannah ask Larry, who last called Betty, and reassured her that he is nice. Hannah preferred if Amanda texted Larry, and Amanda encouraged her to do so.

Ours: Hannah is looking for Betty's number but Amanda can't find it. Larry called her last time they were at the park together.

Golden Summary: Hannah needs Betty's number but Amanda doesn't have it. She needs to contact Larry.

Luke: are you still looking for someone to join netflix family?
Paul: yes, 1 person :)
Luke: i am the one!
Paul: sure, i will send you the login and password on sunday
Luke: ok we can talk tomorrow
Paul: i don't really remember it now
Luke: send me also the bank account details so I can wire you the money every month. Are you paying for this or someone else?
Paul: I do, and I keep track of everyone accessing so you should not expect any bans :D
Luke: easy mate :D you still on holidays with your girl?
Paul: last dinner :(tomorrow we are out
Luke: how long have you been there?
Paul: less than 8 days :/

DeepSeek^{pt}: Luke asks if Paul is still looking for someone to join the Netflix family, to which Paul confirms he is looking for one person. Luke volunteers himself, and Paul agrees to send him the login and password on Sunday. They discuss further details, with Luke asking for bank account details to send monthly payments, and Paul assuring that he manages access and there won't be any bans. They briefly chat about Paul's holiday, which is ending after less than 8 days.

DeepSeek$_b^{pt}$: Luke is joining Paul's Netflix account and will send payment monthly; Paul will provide login details on Sunday.

DeepSeek$_u^{pt}$: Luke will join Paul's Netflix family and receive login details on Sunday, with plans for Luke to wire monthly payments to Paul.

Ours: Paul will send Luke the login and password to Netflix and the bank account details so he can wire him the money every month. Paul is on holidays with his girl for 8 days.

Golden Summary: Paul is going to share his Netflix account with Luke. In exchange Luke is going to contribute to the subscription. Paul will send Luke his bank details. Paul is on vacation with his girlfriend till tomorrow.

Rachel: <file_other>
Rachel: Top 50 Best Films of 2018
Rachel: :)
Janice: Omg, I've watched almost all 50... xDD
Spencer: Hahah, Deadpool 2 also??
Janice: Yep
Spencer: Really??
Janice: My bf forced me to watch it xD
Rachel: Hahah
Janice: It wasn't that bad
Janice: I thought it'd be worse
Rachel: And Avengers? :D
Janice: 2 times
Rachel: Omg
Janice: xP
Rachel: You are the best gf in the world
Rachel: Your bf should appreciate that ;-)
Janice: He does
Janice: x)

DeepSeek^{pt}: Rachel mentions a list of the Top 50 Best Films of 2018, to which Janice replies that she has watched almost all of them. Spencer asks if Janice watched Deadpool 2, which she did, though she admits her boyfriend made her watch it and it wasn't as bad as she expected. Rachel then asks about Avengers, and Janice says she watched it twice. Rachel jokingly commends Janice for being the best girlfriend, suggesting her boyfriend should appreciate her for watching these films with him. Janice confirms that he does appreciate her.

DeepSeek$_b^{pt}$: Rachel discussed the Top 50 Best Films of 2018 with Janice, who had watched almost all of them, including those she was initially hesitant about due to her boyfriend's influence.

DeepSeek$_u^{pt}$: Rachel, Janice, and Spencer discussed the top 50 best films of 2018, with Janice mentioning she watched almost all of them, including Deadpool 2 and Avengers, multiple times.

Ours: Janice has watched almost all 50 best films of 2018. She watched Deadpool 2 and Avengers 2 times.

Golden Summary: Rachel sends a list of Top 50 films of 2018. Janice watched almost half of them, Deadpool 2 and Avengers included.

Alex: Were you able to attend Friday night's basketball game?
Benjamin: I was unable to make it.
Alex: You should have been there. It was intense.
Benjamin: Is that right. Who ended up winning?
Alex: Our team was victorious.
Benjamin: I wish I was free that night. I'm kind of mad that I didn't go.
Alex: It was a great game. Everything alright tough?
Benjamin: Yeah man thanks for asking, it's just that my mom is sick and I am taking care of her.
Alex: Oh sorry to hear that. Hope she makes a fast recovery
Benjamin: She will, she just has a nasty flu but she will be alright :D
Alex: Glad to hear that!
Benjamin: What was the score at the end of the game?
Alex: Our team won 101-98.
Benjamin: Sounds like it was a close game then.
Alex: That's the reason it was such a great game.
Benjamin: I'll go to the next one for sure.
Alex: It's next weekend so you better put on your calendar ahaha
Benjamin: ahaha I will I will. Talk to you later!
Alex: Alright! Tell your mom I hope she gets better quickly.

DeepSeek: Alex and Benjamin discuss a basketball game that took place on Friday night. Benjamin missed the game due to taking care of his sick mother. The game was intense, and their team won with a score of 101-98. Benjamin regrets not attending and plans to go to the next game. Alex wishes Benjamin's mother a speedy recovery.

DeepSeek$_b^{pt}$: Alex and Benjamin discussed missing a basketball game where their team won 101-98, and Benjamin expressed regret due to family obligations.

DeepSeek$_u^{pt}$: Alex discussed the intense basketball game with Benjamin, who missed it due to taking care of his sick mother. The game ended with Alex's team winning 101-98, and Benjamin plans to attend the next game.

Ours: Benjamin wasn't able to attend Friday night's basketball game because his mother is sick. His team won 101-98. Benjamin will go to the next one next weekend.

Golden Summary: Benjamin didn't come to see a basketball game on Friday's night. The team supported by Alex won 101-98. Benjamin's mom has a flu and he's looking after her. Benjamin declares to attend the next basketball match.