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, extensive 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 tackle this challenge, we propose the extract-then-generate methodology. Our rationale behind this approach aligns with the cognitive process observed in human dialogues: selecting **U**tterances **M**ost related to speakers' key **I**ntents (UMIs) and summarizing them (Mao et al., 2022). Given that dialogue summaries commonly revolve around discerning "who did what" (Liu and Chen, 2021), we assert that gathering UMIs scattered throughout the dialogue is instrumental for models to deduce the **K**ey **I**ntents (KIs) of speakers, thereby enhancing the fidelity of generated results. While previous research has delved into techniques that integrate both extraction and summarization components (Lebanoff et al., 2018; Xu and Durrett,

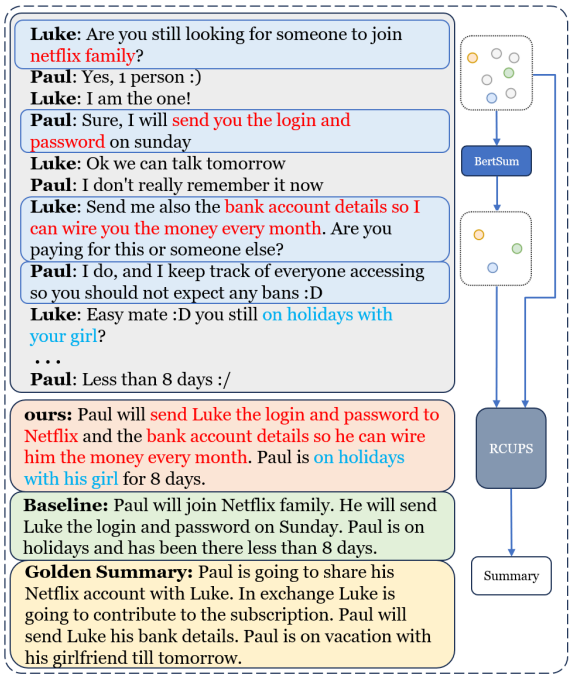

Figure 1: A dialogue summary samples generated by the baseline and the RCUPS model, reveal that the selected utterances effectively manifest the pertinent information in the summary, Meanwhile, RCUPS does not neglect the information in utterances that were not selected. In contrast, the baseline lacks 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-0018

2019; Zhang et al., 2019a; Lebanoff et al., 2019; Zou et al., 2020; Bajaj et al., 2021; Zhang et al., 2021), these models typically follow a sequential connection between the extraction and summarization processes, as depicted in Figure 2, generating summaries based on extracted content. Moreover, alternative approaches, such as those involving entity chains (Narayan et al., 2021) or the utilization of named entity sequences to enhance content control (Liu and Chen, 2021), lack a dedicated focus on capturing the essential intentions of speakers. In contrast, the work by Yoo and Lee (2023) employs keyword extraction while retaining the original text. However, it may fall short in generating contextually coherent summaries due to discrete token combinations. Notably, all these approaches involve a mere concatenation of extracted features with dialogue text, as illustrated in Figure 2.

Therefore, we propose an algorithm designed to select UMIs based on the summary. This approach draws inspiration from the Target Matching methodology (Zhang et al., 2022b). The algorithm operates on two key assumptions: (1) long sentences within a dialogue inherently contain rich and crucial information; (2) sentences in the golden summary exhibit semantic independence, adhering to a "who did what" format, allowing each sentence to serve as a representation of a **K**ey **I**ntent (KI) pertaining to the subject involved. Furthermore, adopting utterance-level matching serves to enhance the accuracy and coherence in representing a dialogue. In our study, we employ BertSUM (Liu, 2019) as a trainable extractive model. Specific training details are elucidated in Section A.2.

The architectural framework of RCUPS is illustrated in Figure 3. The initial dialogue text undergoes processing through three distinct data streams: plain, utterance, and salient. Inspired by the works of Humeau et al. (2019); Yang et al. (2022); Zhang et al. (2022a); Xie et al. (2022a), we adopt a dual-encoder approach, concurrently encoding the salient stream and the other two streams. The integration of the row-column fusion module amplifies information interaction between the plain and salient streams. This design choice empowers the model to concentrate on the KIs within the dialogue while maintaining awareness of the overall dialogue content. During the decoding phase, the model leverages the condensed information from the salient stream through the "extract-utterances" prefix. This guides the model to prioritize attention to KIs. Following this, the utterance weight is applied to diminish the scores of non-UMIs, aiding the model in sieving out redundant information and achieving a more precise summary.

Our main contributions can be summarized as follows:

- We present the RCUPS model, which incorporates a novel two-dimensional fusion during the encoding stage and integrates information enhancement and weight decay mechanisms in the decoding stage. These features empower the model to intensify its focus on key intents in the text while preserving essential contextual information.
- We introduce an algorithm for extracting **U**tterances **M**ost related to speakers' Key **I**ntents (UMIs) by leveraging the **K**ey **I**ntents (KIs) present in the golden summary. This approach proves to be not only efficient but also effective in generating labels for datasets lacking extractive annotations, thereby stimulating advancements in the extractive summarization domain.
- Our extensive experiments conducted on three dialogue summarization datasets show superior results compared to robust baseline models.

2 Related Work

2.1 Dialogue Summarization

Dialogue summarization represents a pivotal research domain, offering the means to distill valuable insights from extensive conversational exchanges. The seminal work by Gliwa et al. (2019) introduced SAMSum, the high-quality, manually annotated dialogue corpus. This resource paved the way for numerous baseline studies, laying the groundwork for subsequent advancements in dialogue summarization. Addressing this challenge, researchers have embraced graph-based strategies, integrating various features such as discourse graph (Chen and Yang, 2021), Heterogeneous Graph incorporating Commonsense Knowledge (Xiachong et al., 2021), coreference graph (Liu et al., 2021b), and static-dynamic graph (Gao et al., 2023) to model dialogue interactions. Moreover, to capture the nuances of dialogue participants, approaches such as named entities planning (Liu and Chen, 2021), speaker-aware self-attention mechanisms

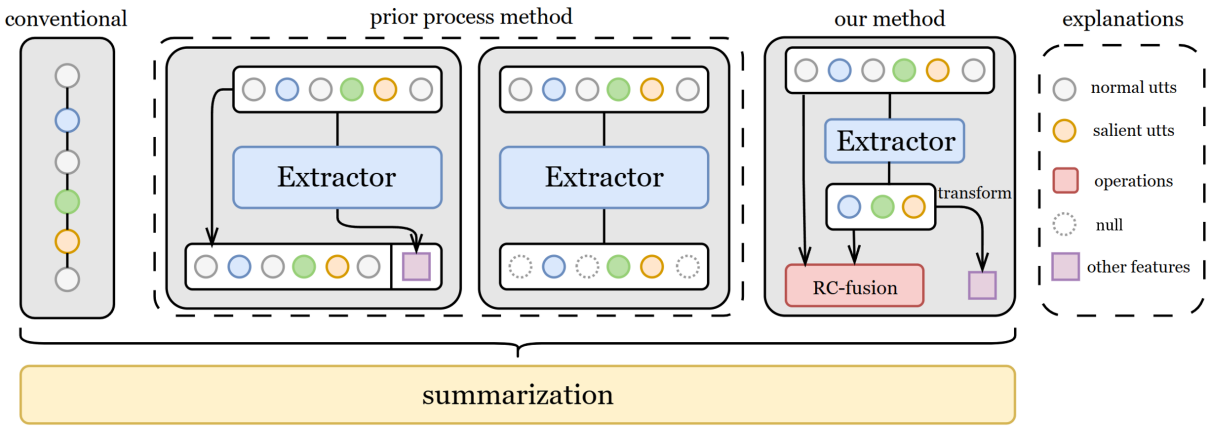

Figure 2: Traditional summarization approaches often resort to a straightforward concatenation of dialogues in chronological order. Meanwhile, prevailing methods in the field typically rely on either exclusively utilizing extracted sentences for generating content or extracting additional information, 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.

(Lei et al., 2021), time-speaker streams (Xie et al., 2022b), and speaker-aware supervised contrastive learning (Geng et al., 2022) have been employed. In the pursuit of enriching dialogue understanding, Feng et al. (2021a) introduced an unsupervised DialoGPT annotator, while Chen et al. (2022) proposed various levels of human feedback. Additionally, Wang et al. (2023) presented an approach for synthesizing query-based summarization triples, contributing to the exploration of additional dimensions of dialogue content.

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 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. (2022b) 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 $\hat{D^{m\prime}} = \{\hat{u_1}, \cdots, \hat{u_{m'}}\}$ denotes a selected subset (UMIs) of D^m and can be obtained with Algorithm 1. $m\prime$ represents the element

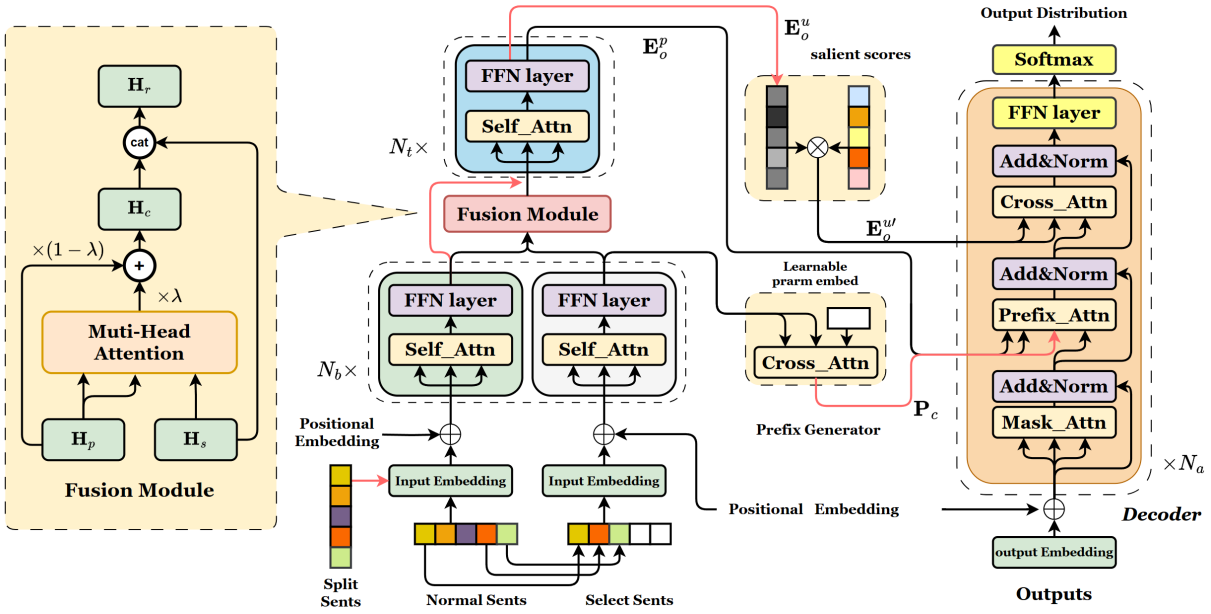

Figure 3: Overview of RCUPS

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.

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 $\mathbf{H}_p$.
- **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.
- **Salient stream**: We use a pre-trained BERT model (Liu, 2019) to extract UMIs, and view

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

all UMIs in a sequence, which we denote as $\mathbf{H}_s$.

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 denoted 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 $\mathbf{E}_o^{w\prime}$ 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 $E_o^{w\prime}$ 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. **MV-BART** (Chen and Yang, 2020) is a methodology derived from BART that integrates both topic and stage data to accurately represent the structure inherent in the dialogue context. **CODS** (Wu et al., 2021) proposes a method for dialogue summarization that allows control over the level of granularity. **BART($\mathcal{D}_{\text{ALL}}$)** (Feng et al., 2021b) uses the DialoGPT (Zhang et al., 2019c) 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. **SCL** (Geng et al., 2022) proposes speaker-aware supervised contrastive learning for better factual consistency. **HITL** (Chen et al., 2022) incorporates human feedback into the training of the summarization model. **ATM** (Xie et al., 2022a) proposes a 2D view of dialogue based on a time-speaker perspective. **SummaFusion** (Ravaut et al., 2022a) fuses several summary candidates to produce a second-stage summary. **SICK++** (Kim et al., 2022) proposes to leverage the unique characteristics of dialogues sharing commonsense knowledge across participants to resolve the difficulties in summarization. **DADS** (Li et al., 2023) proposes to use a disentangled representation method to reduce the deviation among data in different domains.

Method	R-1	R-2	R-L
SAMSum			
Oracle†	57.99	32.01	59.17
CODS	52.65	27.84	50.79
MV-BART	53.42	27.98	49.97
BART($\mathcal{D}_{\text{ALL}}$)	53.70	28.79	50.81
CONDIGSUM	54.30	29.30	45.20
Coref-Attn	53.93	28.58	50.39
SCL	54.22	29.87	51.35
HITL	53.76	28.04	50.56
SummaFusion	52.76	28.24	43.98
SICK++	53.73	28.81	49.50
DADS	54.22	29.04	51.08
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.

4.2 Evaluation Metrics and Datasets

For evaluation metrics, following existing dialogue summarization papers (Feng et al., 2021a), we adopt ROUGE score (Lin, 2004) to assess the quality of generated summaries, which consider the overlapping uni-grams, bi-grams, and the longest common subsequences, respectively. Due to the potential for misdirection when only using automatic evaluation metrics (Stent et al., 2005), we

also employ evaluation methods based on embeddings and conduct the human evaluation. We use BERTScore (Zhang et al., 2019b) and BARTScore (Yuan et al., 2021) as our embedding-based evaluations. Datasets statistics can be found in 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	BERTScore	BARTScore
BART$_{large}$	91.67	-2.33
RCUPS	**92.86**	**-2.27**

Table 2: Semantic similarity evaluation on SAMSum.

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 congruence 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 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. 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 3. 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
Human-written	4.71	4.65	4.38
RCUPS	4.40	4.61	4.10

Table 3: human evaluation result. **Fai.** for Faithfulness. **Flu.** for Fluency. **Inf.** for Informativeness

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			
-λ = 0.6	54.49	29.35	55.63
-λ = 0.7	54.53	29.49	55.72
-λ = 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 investigate the impact of various modifications to the fusion module, including the addition of two streams and the removal of either the row or column part. Additionally, we explore the effects of different λ values on the fusion module, as detailed in Table 4. Notably, our findings indicate that a simple addition of streams does not yield significant improvements in the model's performance.

The removal of either the row or column part results in a deterioration in model performance, with the column part showing a more significant contribution to the decline. This observation underscores the importance of the fusion method in influencing the model's comprehension and generation capabilities. Regarding the influence of λ, we observe that as λ increases, the ROUGE-1 scores initially rise before subsequently declining. This suggests that the encoder benefits from a balanced information fusion module to effectively integrate the two streams, thereby achieving optimal performance.

Effect of Salient Scores Our experiments also investigate the impact of salient scores on the model's performance. As depicted in Table 4, the ROUGE scores demonstrate a decline when the Salient Scores are removed. Furthermore, neglecting the softmax function leads to a significant decrease in scores. We attribute this phenomenon to two main factors: (1) The non-UMIs, although not directly related to the key intents, still carry a small amount of contextual information that contributes to the generation of a more coherent summary. (2) Simply zeroing out these vector representations may confuse the model and potentially trigger a collapse in performance. Hence, the softmax operation is deemed vital, as it ensures the proper functioning of the second decoding phase and maintains the balance between salient and non-salient information, thereby enhancing the overall quality of the generated summaries.

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 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.

7 Conclusion

To enable our model to fully concentrate on salient information and effectively capture the key intentions of dialogue, we introduce an extractive method for generating training labels, coupled with the Row-Column Fusion Dual-Encoders and Utterance Prefix for Dialogue Summarization (RCUPS) method. RCUPS comprises two integral modules: a row-column fusion module at the Encoder and a salient utterance prefix module at the Decoder. The row-column fusion module facilitates the injection of salient information into the summarization model, enhancing its ability to capture essential details during encoding. Meanwhile, the salient utterance prefix module enriches salient information to aid the model in decoding and generating concise summaries. Empirical results demonstrate that the proposed RCUPS method yields significant improvements compared to robust baseline models across three prominent dialogue summarization datasets: SAMSum, DialogSum, and TODSum.

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. 2022. Human-in-the-loop abstractive dialogue summarization. *arXiv preprint arXiv:2212.09750*.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. *arXiv preprint arXiv:2010.01672*.

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. *arXiv preprint arXiv:2209.00930*.

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*.

Jinpeng Li, Yingce Xia, Xin Cheng, Dongyan Zhao, and Rui Yan. 2023. Learning disentangled representation via domain adaptation for dialogue summarization. In *Proceedings of the ACM Web Conference 2023*, pages 1693–1702.

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. *arXiv preprint arXiv:2109.04994*.

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. *arXiv preprint arXiv:1908.08345*.

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 F Chen. 2021b. Coreference-aware dialogue summarization. *arXiv preprint arXiv:2106.08556*.

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 Joty, and Nancy F Chen. 2022a. Towards summary candidates fusion. *arXiv preprint arXiv:2210.08779*.

Mathieu Ravaut, Shafiq R. Joty, and Nancy F. Chen. 2022b. 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.

Chien-Sheng Wu, Linqing Liu, Wenhao Liu, Pontus Stenetorp, and Caiming Xiong. 2021. Controllable abstractive dialogue summarization with sketch supervision. *arXiv preprint arXiv:2105.14064*.

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. 2019c. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

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: $T = \{t_1, t_2, \cdots, t_n\}, U = \{u_1, u_2, \cdots, u_m\}$

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.