RepSum: Unsupervised Dialogue Summarization based on Replacement Strategy

Xiyan Fu[1], **Yating Zhang**[2], **Tianyi Wang**[2]
Xiaozhong Liu[3], **Changlong Sun**[2] and **Zhenglu Yang**[1]
[1] Nankai University, Tianjin, China
[2] Alibaba Group, Hangzhou, China
[3] Indiana University, Bloomington, USA
fuxiyan@mail.nankai.edu.cn, {ranran.zyt,will.wty}@alibaba-inc.com
liu237@indiana.edu, changlong.scl@taobao.com, yangzl@nankai.edu.cn

Abstract

In the field of dialogue summarization, due to the lack of training data, it is often difficult for supervised summary generation methods to learn vital information from dialogue context. Several works on unsupervised summarization for document by leveraging semantic information solely or auto-encoder strategy (i.e., sentence compression), they however cannot be adapted to the dialogue scene due to the limited words in utterances and huge gap between the dialogue and its summary. In this study, we propose a novel unsupervised strategy to address this challenge, which roots from the hypothetical foundation that a superior summary approximates a replacement of the original dialogue, and they are roughly equivalent for auxiliary (self-supervised) tasks, e.g., dialogue generation. The proposed strategy *RepSum* is applied to generate both extractive and abstractive summary with the guidance of the followed n^{th} utterance generation and classification tasks. Extensive experiments on various datasets demonstrate the superiority of the proposed model compared with other unsupervised methods.

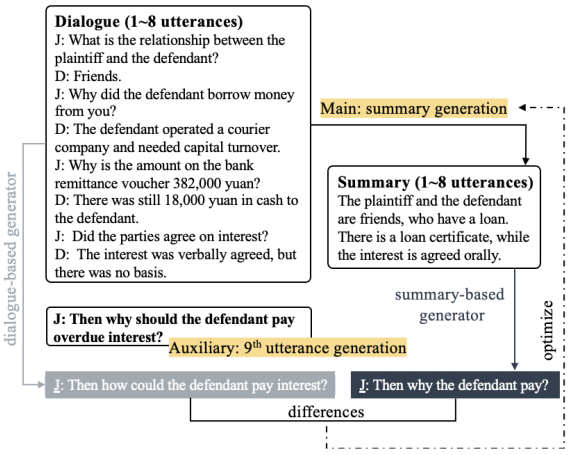

Figure 1: A summary is generated from the input dialogue firstly, and then the original dialogue and its corresponding summary are exploited for n^{th} utterance prediction, respectively. J is the ground truth, and J in different colors are the decoded utterances based on the original dialogue and the generated summary respectively. The difference between decoded J is employed for optimization of summary generation. The motivation is that a superior summary approximates a replacement of the original dialogue, and they are roughly equivalent for auxiliary tasks.

1 Introduction

Dialogue summarization distills key information from a dialogue context and synopsizes it into a concise summary. As a novel topic of critical importance, it offers powerful potentials for a number of scenarios, e.g, the court debate in civil trial, the customer service calls arisen from agent(s) and customer, the business meeting engaged with multi-members. It also assists users in quick access and consumes the essential content in the dialogue.

Major attempts on dialogue summarization are template-based (Wang and Cardie, 2013; Oya et al., 2014) in the primitive stage by extracting key information and filling it into the learned templates. However, these template-based techniques limit the scope of their applications and cannot be adapted to a wider range of conversational data since their input structure is predefined and the learned templates are domain-specific. Later, various works explore the assistance from labeled auxiliary information for summary generation, by leveraging either dialogue act (Goo and Chen, 2018), or key point sequence (Liu et al., 2019). The former predicts the dialogue act label of each utterance as explicit interactive signals, while the latter attempts to learn the logic of the summary via key point sequence. Recently, Ganesh and Dingliwal (2019) converts the dialogue into a document by aptly capturing discourse relations which proves to be effective under the scenario of document summarization.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 6042–6051
August 1–6, 2021.

While prior deep content generation methods rely on large amounts of annotated data, they are rarely available for dialogue summarization due to the prohibitive costs of labeled data. A straightforward way to alleviate the dependency of the annotated data is to apply the existing unsupervised methods designed for document summarization (Rossiello et al., 2017; Zheng and Lapata, 2019; Baziotis et al., 2019; Chu and Liu, 2019) to the dialogue scene. However, we argue that these methods accompany weakness either in extractive or in abstractive dialogue summarization. In terms of extractive methods, they mainly rely on semantic information without any supervision signals. As a result, they are ragged in effects due to the limited words in dialogue utterances. As for abstractive approaches, they are commonly designed with an auto-encoder (AE) where the latent variable decodes to a summary which attempts to reconstruct the original input representation. Hence, they are constrained to the small gap between the input text and the target summary (e.g., sentence compression) while failing to reconstruct long input text (e.g., dialogue).

In this paper, we propose an innovative unsupervised strategy, dubbed RepSum, which can be applied to both extractive and abstractive summarization. The key intuition is derived from the evaluation methods of extrinsic summarization (Mani, 2001), which testifies the impact of summarization based on how it affects the completion of some other tasks, such as information retrieval, relevance assessment, reading comprehension, etc. We claim that **a superior summary can offer a semantic replacement of the original dialogue, which provides equivalent information for completing auxiliary tasks**, e.g., dialogue generation, as shown in figure 1. Specifically, we propose two auxiliary tasks which are n^{th} utterance generation and n^{th} utterance selection from K candidates based on the previous contents. Both the dialogue and the summary aim to achieve decent performances on the specific task respectively. Besides, we introduce KL divergence to curtail the difference between results based on the dialogue and the summary. This strategy provides the summarization with essential self-supervised signals via auxiliary tasks. Furthermore, it decouples the training from the reconstruction of AE, which enables to support longer text or dialogue to be effectively summarized.

Our main contributions are as follows:

- We propose RepSum, an unsupervised (or self-supervised) strategy for dialogue summarization, which roots from the hypothesis that a superior summary approximates a replacement of the original dialogue for completing other tasks. It leverages several intrinsic self-supervised signals.
- Based on the RepSum strategy, we propose the corresponding model and employ it to both extractive and abstractive summarization.
- The extensive experiments with multiple dialogue datasets demonstrate the superiority of the proposed model over several unsupervised approaches.

2 Related Work

Dialogue Summarization extracts significant information from dialogues. Most of the initial works adopted extractive-based methods. For instance, Bui et al. (2009) produced multiple short fragments from utterances and then selected the parse of the summary by SVM combined with semantic-similarity features. Later, (Oya et al., 2014; Wang and Cardie, 2013) induced abstractive generation templates for constructing candidate summary sentences. Moreover, to benefit from the existing technologies for document summarization, Ganesh and Dingliwal (2019) converted the conversation into a text document through discourse relations and lexical information and then created summaries via pointer-generator (See et al., 2017). However, given that dialogues are different from documents in terms of interactive patterns, most researchers explored to summarize the dialogue by leveraging auxiliary information hidden in the utterances. For example, Goo and Chen (2018) proposed to utilize dialogue act as an auxiliary supervised signal and design a sentence-gated mechanism for modeling the relationships between dialogue acts and the summary. In addition, Liu et al. (2019) predicted the keypoint sequence first and then use it to guide the summary prediction.

In contrast to the supervision works, we focus on the unsupervised dialogue summarization considering the high cost and limitation of the labeled data in the dialogue scene. Additionally, our proposed strategy is applicable to both extractive and abstractive models without using any outer infor-

mation (e.g., template, dialogue acts, and keypoint) but leveraging its intrinsic self-supervised nature.

Unsupervised Summarization Historically, unsupervised summarization focused on extracting utterances directly. For example, LEAD chooses the first several utterances and TextRank (Mihalcea and Tarau, 2004) ranks utterances by running a graph-based algorithm, where each node represents an utterance and the weight between any two nodes is calculated by the semantic similarity. Later, Rossiello et al. (2017) proposed a centroid-based method for text summarization that exploits the compositional capabilities of word embeddings. Zheng and Lapata (2019) improved it by building graphs with directed edges considering the relative positions of any two sentences which contributes to their respective centrality. In recent works, the task of unsupervised summarization is framed as a self-supervised auto-encoder problem, namely sentence compression. Miao and Blunsom (2016); Baziotis et al. (2019); Chu and Liu (2019) applied the auto-encoder framework, where the expected abstract is set to the latent variables from which the input sentence is reconstructed. Févry and Phang (2018) added noise to extend sentences and trained a denoising auto-encoder to recover the input text. Bražinskas et al. (2020) introduced a hierarchical variational auto-encoder to associate the individual reviews with stochastic latent codes for opinion summarization. Recently, another line of works focused on edit-distance-based approaches. West et al. (2019) summarized by applying the Information Bottleneck principle to the objective of conditional language modeling. In addition, Zhou and Rush (2019); Schumann et al. (2020) summarized by hill climbing with word-level extraction, which searches the text for a high-scoring summary by discrete optimization.

Compared to these works, to the best of our knowledge, our model is one of the pioneers attempting unsupervised dialogue summarization. To improve the effectiveness, we devise a generalized strategy RepSum that incentivizes the summary to complete the auxiliary tasks as the original dialogue does, thus providing self-training signals and in turn enabling long texts to be summarized.

3 RepSum Model

3.1 Mechanism

RepSum roots from the hypothetical foundation that a superior summary approximates a replacement of the original dialogue, and they are roughly equivalent for completing auxiliary (self-supervised) tasks. Figure 2 shows the flow chart of the introduced replacement strategy. Specifically, the summary generation module aims at generating a summary from the original dialogue. During this generation process, two auxiliary tasks, n^{th} utterance generation and n^{th} utterance classification, are constructed to transform unsupervised dialogue summarization task into self-supervised mode by learning through auxiliary tasks. Furthermore, we apply RepSum to extractive and abstractive summarization, experiments verify its effectiveness in an empirical point of view.

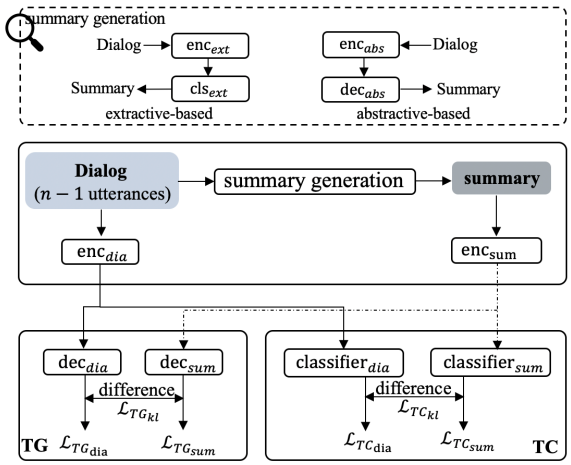

Figure 2: The overall flow chart of the proposed model. The middle square is the unsupervised dialogue summarization generation process. Further, both the dialogue and the corresponding summary are employed on auxiliary tasks (i.e., n^{th} utterance generation and classification). The innovation lies to a superior summary is the replacement of the original dialogue.

3.2 Auxiliary Tasks

As introduced above, we leverage two auxiliary tasks to act as self-supervised signals to assist the generation process of a superior summary. Given that the summary is the replacement of the original dialogue, the input dialogue and the generated summary are expected to achieve similar results on these tasks respectively. Hence, we add the Kullback–Leibler (KL) divergence to curtail the differences between the results of each auxiliary task based on the input dialogue and the generated summary. The details are denoted as follows:

Task1: Generation (TG) aims at generating the n^{th} utterance. We employ the commonly used encoder-decoder structure. The whole dialogue is

concatenated and encoded (as a document) by the bi-directional LSTM(Hochreiter and Schmidhuber, 1997) for the sake of fair comparison with other baselines. The representation of each word is the concatenation of the forward and backward LSTM states, i.e., $h_i = [h_i^{fwd}, h_i^{bwd}]$. As for the decoder, we employ a uni-directional LSTM with attention mechanism (Luong et al., 2015). Concretely, the attention distribution a^t and the following context vector c_t are formulated as:

$$a_i^t = \sigma(h_i W_a s_t), \quad c_t = \sum_{i=1}^{n} a_i h_i \tag{1}$$

where W_a is the learnable parameter and σ is the softmax function. The context vector and the current decoder state s_t are employed for predicting the probability distribution of the output word over all the vocabulary words:

$$p(y_t) = \sigma(W_p(\phi(W_k[y_{t-1}; s_t; c_t] + b_k)) + b_p) \tag{2}$$

where W_p, W_k, b_p and b_k are learnable parameters. σ is the softmax function and ϕ is the tanh function. We choose the negative logliklihood as the loss function, and the loss of the utterance generation based on the dialogue via the path of $enc_{dia} \rightarrow dec_{dia}$ (see Figure 2) is denoted as:

$$\mathcal{L}_{TG_{dia}} = -\sum_{t=1}^{q} \log p(l_t | l_{<t}; enc_{dia}) \tag{3}$$

where $l = \{l_1, l_2, ..., l_q\}$ is the generated utterance. Similarly, the utterance generation based on the generated summary $\mathcal{L}_{TG_{sum}}$ is calculated via the process of $enc_{sum} \rightarrow dec_{sum}$ in Figure 2. To guarantee the similar performance of the results based on the original dialogue and the generated summary, we also add KL divergence to curtail the difference between the probability distribution of prediction at each timestep:

$$\mathcal{L}_{TG_{kl}} = \sum_{t=1}^{q} KL(p(l_t | l_{<t}; enc_{dia}) || p(l_t | l_{<t}; enc_{sum})) \tag{4}$$

Hence, the loss for the n^{th} utterance generation task is denoted as:

$$\mathcal{L}_{TG} = \alpha_0 \mathcal{L}_{TG_{dia}} + \alpha_1 \mathcal{L}_{TG_{sum}} + \alpha_2 \mathcal{L}_{TG_{kl}} \tag{5}$$

where α_0, α_1, and α_2 are the weight for each loss.

Task2: Classification (TC) is designed to select the correct n^{th} utterance from the K candidate utterances. Similar to the dialogue encoding in the task TG, we choose the Bi-LSTM as the encoder. The dialogue representation h_d is the average of the hidden state of each word. Besides, each candidate is also encoded by the Bi-LSTM and projected to a dense vector by logit layer f, and then concatenated to h_d, formulated as $[f(uc_i); h_d]$. The probability of each utterance belonging to the correct answer is calculated by a logistic layer. Furthermore, we use cross-entropy for training via the process of $enc_{dia} \rightarrow classifier_{dia}$ (see Figure 2). The loss based on the dialogue is formulated as:

$$\mathcal{L}_{TC_{dia}} = -\sum_{n=1}^{K} z_n log \hat{z}_n \tag{6}$$

Similarily, $\mathcal{L}_{TC_{sum}}$ based on the generated summary through $enc_{sum} \rightarrow classifier_{sum}$ is calculated. We also use the KL divergence to measure the difference between the results from the dialogue and generated summary:

$$\mathcal{L}_{TC_{kl}} = KL(p(uc^{dia}) || p(uc^{sum})) \tag{7}$$

where $p(uc^{dia})$ and $p(uc^{sum})$ is the probability distribution on K candidates. All in all, the loss of the n^{th} utterance selection task is formulated as:

$$\mathcal{L}_{TC} = \alpha_3 \mathcal{L}_{TC_{dia}} + \alpha_4 \mathcal{L}_{TC_{sum}} + \alpha_5 \mathcal{L}_{TC_{kl}} \tag{8}$$

where α_3, α_4, and α_5 are the weight for each loss. Parameters α_0 to α_5 are used for normalization.

3.3 Unsupervised Summarization

The RepSum is employed to both the extractive and abstractive summarization:

Extractive Summarization We consider the extractive summarization as a sentence binary classification task as (Nallapati et al., 2017) does, which means R utterances in a dialogue with label one are extracted to be an extractive summary. Specifically, we use enc_{ext} (enc_{dia} in the Figure 2) applied by the Bi-LSTM to encode utterances in dialogue, and they are represented as hidden states $h_1, h_2, ..., h_{n-1}$. Then, the representation of the dialogue is the average pooling of the concatenated hidden states of the entire utterances, denoted as:

$$d = \phi(W_d \frac{1}{n-1} \sum_{i=1}^{n-1} [h_i^{fwd}; h_i^{bwd}] + b_d) \tag{9}$$

where W_d and b_d are learnable parameters, and ϕ is the tanh function. For utterances classification,

each utterance is concatenated with the dialogue representation d. And a logistic layer predicts the probability belonging to the generated summary, as shown below:

$$p(u_i = 1) = \psi(W_h h_i + h_i W_{hd} d + b_h) \quad (10)$$

where W_h, W_{hd} and b_h are learnable parameters, and ψ is the sigmoid function. Later, we choose the top probability R utterances as the extractive summary. After obtaining the initial generated summary, the unsupervised extractive summarization can be guided under the RepSum strategy. Specifically, the extractive-based summary is optimized by the auxiliary tasks for the sake of effective results and similar performance of the dialogue. Hence, the training loss for extractive summarization including n_{th} utterance generation and classification is denoted as:

$$\mathcal{L}^{ext} = \mathcal{L}^{ext}_{TG} + \mathcal{L}^{ext}_{TC} \quad (11)$$

Abstractive Summarization The abstractive summarization process follows the conventional encoder-decoder structure. For each time step, the word prediction probability is calculated via Eq. 2. To generate the abstractive summary used for the auxiliary tasks, we sample each word from the probability $\widetilde{y}_t \sim softmax(p(y_t))$ and encode them as enc_{a_sum} (enc_{sum} in the Figure 2). However, it is a non-differentiable process, which can not be trained directly.

Hence, we use the Straight-Through (ST) Gumble Estimator introduced in (Bengio et al., 2013) to solve this problem. During the forward training pass and test process, we use the reparametrization trick as a variance approximation of sampling from the original probability (Maddison et al., 2014). Specifically, sampling word is transformed to take the $argmax$ from a new probability, $\widetilde{y}$ is discretized using $argmax$ and sampling as:

$$\begin{aligned} \widetilde{y}_t &= argmax(log(p(y_t)) + g), \\ g &= -log(-log(\xi)), \quad \xi \sim U(0,1) \end{aligned} \quad (12)$$

where g is the Gumble distribution and U is the uniform distribution. As for computing the gradient in the backward pass, we use a continuous and differentiable approximation to $argmax$:

$$p(y^i_t) = \frac{exp((log(p(y^i_t)) + g_i)/\tau)}{\sum_{j=1}^{|V|} exp((log(p(y^j_t)) + g_j)/\tau)} \quad (13)$$

	AMI	Justice
Test Set Size	132	1525
Avg. Utterance Num	219.89	37.54
Avg. Utterance Length	293.26	16.11
Avg. Summary Length	161.33	159.50

Table 1: Statistics of datasets

where $|V|$ is the vocabulary size and the $\tau \in (0, \infty)$ is the temperature parameter. Samples from Gumble Softmax distributions are identical to samples from a categorical distribution as $\tau \to 0$. The input for the encoder enc_{a_sum} is denoted as:

$$e^{abs}_{y_t} = \sum_{i=1}^{|V|} e(w_i) p(y^i_t) \quad (14)$$

where $e(w_i)$ is the word embedding of the words. After the acquisition of the abstractive summary, we also employ the RepSum strategy for training. Due to the difficulty of the generation, we supply two more other auxiliary losses. Firstly, the experiments indicate that the model is difficult to converge due to the lack of any guidance for the decoder (see w/o fake-sum in Table 5), we employ the extractive summary as a fake summary for teacher forcing training. Hence, the fake summary generation loss $\mathcal{L}_{fs}$ is calculated following the Eq. 1, Eq. 2 and Eq. 3. Moreover, given that abstractive summary is limited to readability and fluency, we pre-train a language model with dialogue utterances to solve this problem. We aim to generate fluent summaries by adding language modeling loss, which approaches the output prediction to language output:

$$\mathcal{L}_{lm} = KL(p(y_t)||p_{lm}(y_t)) \quad (15)$$

Hence, the training loss for the unsupervised abstractive dialogue summarization is denoted as:

$$\mathcal{L}_{abs} = \mathcal{L}^{abs}_{TG} + \mathcal{L}^{abs}_{TC} + \alpha_6 \mathcal{L}_{fs} + \alpha_7 \mathcal{L}_{lm} \quad (16)$$

Parameters α_6 and α_7 are normalization weight.

4 Experimental Setup

4.1 Dataset

We evaluate RepSum on a meeting dataset in English **AMI** and a multi-party court debate dataset in Chinese **Justice**. The statistics are presented in details (see Tabel 1).

AMI. The AMI[1] meeting corpus (Carletta et al., 2005) consists of 100 hours of meeting recordings

[1] http://groups.inf.ed.ac.uk/ami/corpus/overview.shtml

Type	Model	AMI			Justice		
		R-1	R-2	R-L	R-1	R-2	R-L
Extractive	ORACLE	24.57	4.44	15.03	37.28	21.05	32.78
	LEAD3	9.15	1.78	5.36	17.69	3.33	11.52
	TextRank (Mihalcea and Tarau, 2004)	11.27	0.84	7.19	20.72	6.51	13.56
	Centroid (Rossiello et al., 2017)	14.08	2.09	8.19	22.31	6.53	13.66
	PacSum(Zheng and Lapata, 2019)	16.15	2.23	9.14	23.36	7.03	14.66
	RepSum-Ext (ours)	**18.77**	**2.24**	**10.80**	**25.88**	**8.21**	**15.97**
Abstractive	2g shuf Févry and Phang (2018)	14.08	2.09	8.18	20.19	4.15	12.08
	MeanSum(Chu and Liu, 2019)	16.09	2.30	11.14	21.25	5.54	13.44
	SEQ^3(Baziotis et al., 2019)	17.06	2.23	11.85	22.47	3.88	14.67
	RepSum-Abs (ours)	**18.88**	**2.38**	**15.62**	**24.23**	**6.37**	**15.14**

Table 2: Comparison of our mechanism employed in extractive and abstractive summarization with other baseline models. All the results are evaluated by the ROUGE on the AMI nad Justice dataset (pairwise t-test at 5% significance level).

in English. It includes high-quality and manually produced transcription, dialogue acts, topic segmentation, extractive and abstractive summaries, etc. In this work, we use the recording transcripts as the original input and the provided abstractive summary as the expected summary to be generated.

Justice. The court debate records consist of 30,000 dispute cases. In the court trial scenario, there are multiple roles (i.e., judge, plaintiff, defendant). In the whole debate dialogue, the plaintiff and the defendant debate on controversy focus leading by the judge. After the trial, the judge summarizes the facts recognized through the trial. Thus we use the court debate transcript as the original input and the fact description as the expected summary.

4.2 Parameter Settings

In our experiments[2], we optimize the proposed model using Adam Optimizer (Kingma and Ba, 2014) with the learning rate of 3e-4. We train on a single TeslaP100 GPU with a batch size of 16. The vocabulary size is 30,000 and embedding dimension for each word is 200. The hidden size is 200 for both encoder and decoder. For gumble softmax, we set the temperature τ to 0.5. In the auxiliary task C_2, we denote k as 4, which means we select the other 3 similar utterances. They are chosen from all the utterances in the dataset randomly. For extractive summarization, we pick out the top 3 utterances by their probability. We set the $\alpha 0$ to $\alpha 7$ equals 0.5, 0.5, 5, 1, 1, 2, 1, 0.006 respectively to balance the scale of each module.

4.3 Baselines

We firstly report the performance of the *ORACLE* as an upper bound, which uses a greedy algorithm to extract several utterances to maximize the ROUGE compared with the ground truth. *LEAD3* extracts the first three utterances as the summary.

As for the extractive-based methods, we compare with classical *TextRank* (Mihalcea and Tarau, 2004) which converts the dialogue to a weighted-graph where each node represents an utterance and the edge weight expresses the semantic similarity between any two utterances. *Centroid* (Rossiello et al., 2017) proposes a centroid-based method for text summarization that exploits the compositional capabilities of word embeddings. *PacSum* (Zheng and Lapata, 2019) improves the TextRank by building graphs with directed edges considering the relative positions of any two sentences contributing to their respective centrality.

With regard to the abstractive-based methods, we compare with several auto-encoder based approaches. *2g shuf* (Févry and Phang, 2018) adds noise to extend sentences and trains a denoising auto-encoder to recover the original input text. *SEQ^3* (Baziotis et al., 2019) constructs a compressor to generate summary and a reconstructor to regenerate input sentence via two chained encoder-decoder pairs. *MeanSum* (Chu and Liu, 2019) employs the mean of the representations of the input to decode a reasonable summary.

5 Experimental Results

5.1 Quantitative Analysis

Table 2 shows the experimental results based on the AMI and the Justice datasets. ROUGE [3] score (Lin, 2004) is used for evaluation.

For extractive summarization, we found the upper bound ORACLE is quite low in dialogue summarization (see the first row in Table 2) compared

[2]The code can be found in https://github.com/xiyan524/RepSum

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

Model	AMI				Justice			
	Relevance		Fluency		Relevance		Fluency	
	Avg	κ	Avg	κ	Avg	κ	Avg	κ
TextRank	0.57	0.51	1.55	0.81	0.69	0.68	1.34	0.76
Centroid	0.88	0.83	1.64	0.80	1.15	0.71	1.42	0.81
PacSum	1.02	0.77	1.67	0.76	1.13	0.66	1.51	0.79
RepSum-Ext	**1.17**	0.79	**1.69**	0.81	**1.21**	0.63	**1.54**	0.76
2g shuf	0.56	0.78	0.78	0.76	0.71	0.63	0.81	0.81
MeanSum	0.89	0.84	0.89	0.68	0.83	0.61	1.02	0.67
SEQ^3	1.11	0.81	1.03	0.69	1.09	0.59	1.18	0.72
RepSum-Abs	**1.23**	0.82	**1.22**	0.72	**1.17**	0.68	**1.20**	0.69

Table 3: Human evaluation. We report the average score (Avg) and the κ value in relevance and fluency.

Type	Task	AMI			Justice		
		R-1	R-2	R-L	R-1	R-2	R-L
Ext.	TG+TC	18.77	2.24	10.80	25.88	8.21	15.97
	-w/o TC	18.36	2.18	9.94	25.45	7.89	15.51
	-w/o TG	16.89	2.11	9.26	22.80	6.33	14.24
Abs.	TG+TC	18.88	2.38	15.62	24.23	6.37	15.14
	-w/o TC	18.60	1.94	10.55	23.63	6.51	14.29
	-w/o TG	16.13	1.72	10.05	22.75	5.20	13.50

Table 4: Ablation study for the auxiliary tasks in replacement mechanism on the AMI and Justice dataset. Ext. and Abs. represent extractive and abstractive based summarization respectively.

mothods	R-1	R-2	R-L
ResSum-Abs	18.88	2.38	15.62
-w/o dia-task	16.55	1.11	13.49
-w/o sum-task	14.34	1.31	9.78
-w/o kl	16.77	2.20	14.79
-w/o lm	17.87	0.70	13.37
-w/o fake-sum	-	-	-

Table 5: Ablation study of each component based on abstractive summarization on the AMI dataset.

with the document summarization where R-1 score usually approaches to 50 as reported in (Liu and Lapata, 2019). It indicates that the dialogue summarization is much more challenging. Additionally AMI dataset is more appropriate for abstractive summarization since its ORACLE scores are much lower than those for Justice dataset. The score of LEAD3 estimates the information distribution over dialogues. Furthermore, our proposed RepSum-Ext is compared with other four state-of-the-art models with significant improvement in Rouge score. Table 2 demonstrates that the RepSum strategy is effective for extractive summarization.

For abstractive summarization, we mainly compare RepSum-Abs with AE-based methods. We employ the same encoder and decoder settings for baselines for a fair comparison. In terms of ROUGE value, our model outperforms all the baselines, especially in R-L score. We consider that the auxiliary tasks training mechanism helps to prevent the focus on single-word reconstruction, but aims to remain significant continuous information.

5.2 Human Evaluation

In order to ensure the rationality/correctness of the generated summary, we also conducted a human evaluation. The annotators are required to estimate the quality of the generated summaries with respect to the *relevance* indicating the connection between the dialogue and the summary and *fluency* representing the readability. The scores are divided into three levels: +2, +1, 0, in which a higher score stands for excellent. We report the average score and coefficient κ which indicates the consistency of evaluation by different annotators. Specifically, we choose 100 examples for each dataset and six annotators are required to evaluate all the tested methods. The annotators are experienced graduate students who have taken the annotation training before the experiment. Results shown in Table 3 indicate that our proposed strategy is superior to all the baselines. Furthermore, compared to the abstractive-based methods, extractive-based methods perform better on fluency. We consider that the difference is due to sentence integrity.

5.3 Ablation Study

To evaluate the effectiveness of the proposed RepSum strategy, we conduct two ablation studies. We first measure the influence of each auxiliary task (see Table 4). Further, we verify the contribution of each module, shown in Table 5.

Table 4 indicates that combining the two auxiliary tasks achieves the best performance on both extractive and abstractive methods. The decline of performance is observed once we remove either task, especially the generation task. We assume that the classification task is considerably straightforward, which may not require affluent semantic information. However, it serves as an auxiliary section with complicated generation tasks.

Furthermore, we remove each component to investigate the module effectiveness in RepSum-Abs. The result is shown in Table 5. It indicates that all the components make a positive contribution. To be specific, fake summary (-w/o fake-sum) is the critical point, which contributes to the model convergence. Besides, if we remove tasks based on the generated summary (-w/o sum-task), the performance declines significantly. It proves the assumption that a superior summary is supposed to conduct the auxiliary tasks as original dialogue does. Either removing tasks based on the dialogue (-w/o dia-

Fake summary	R-1	R-2	R-L
random	15.45	2.39	10.07
extractive-based	18.88	2.38	15.62

Table 6: Effectiveness of potential fake summary choices for abstractive summarization on the AMI.

T	AMI			Justice		
	R-1	R-2	R-L	R-1	R-2	R-L
3	10.38	0.88	7.13	15.71	3.03	9.92
4	19.87	2.20	11.05	22.80	6.33	14.24
5	18.63	1.85	10.73	22.15	5.52	13.63
6	18.65	1.94	10.61	22.67	6.06	13.98
7	18.77	1.84	10.72	22.51	5.87	13.87

Table 7: Effectiveness of candidate numbers in the auxiliary task classification. It is based on the extractive summarization of the AMI and Justice dataset.

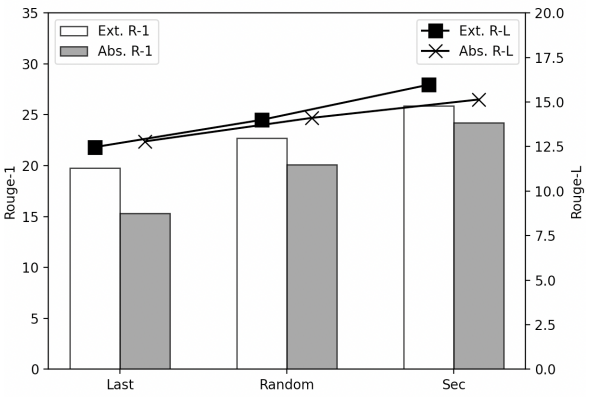

Figure 3: Effectiveness of n^{th} utterance selection in the auxiliary task generation. It is based on the Justice dataset.

task) or adding KL divergence (-w/o kl) to control similar effectiveness between dialogue and generated summary, tends to harm the performance. Moreover, we notice that the pre-trained language model (-w/o lm) benefits the bi-gram by noticing the significant decrease in R-2. The extractive-based method is ignored since its components are the same as the abstractive-based approach.

5.4 Discussion

Fake Summary Extensive experiments show that abstractive summarization is difficult to converge without word-level guidance. Hence, we propose to construct a fake summary to solve this problem. In this section, we conduct two experiments for different fake summary construction. We first attempt to select T utterances randomly. Further, we choose an extractive summary. Table 6 shows that the random selection result is inferior to extractive summary guidance. Given the consideration of high accuracy, we choose the extractive summary as guidance in this work. However, we assume that random selection can be also employed for efficiency consideration if necessary.

Candidates number in TC To further explore the effectiveness of the auxiliary task classification (TC) for unsupervised dialogue summarization, we conduct experiments by varying the candidate's number K. Such number influences the performance of the extractive summarization on both AMI and Justice datasets. We set the number varying from 3 to 7. The performance of our model with the variation of the number K is shown in Table 7. It indicates that the R-1 approaches a stable value with slight fluctuation when we increase the K continuously. Besides, there exists a drastic increase in R-1 when K is augmented from 2 to 3. Hence, given the trade-off between the efficiency and the generation quality, we choose 4 as the number of candidates for all the experiments.

Utterance choice in TG The selection of n^{th} utterance for generation in the dialogue is crucial for the model effectiveness. Meaningless utterances such as "hmmm", "the meeting is over" in meeting, and "please sign the transcript after checking" in court debates may be useless. At the same time, none of the contextual information is integrant. Hence, we conduct experiments to testify the effectiveness of three different utterance selection strategies: *Random* selects the n^{th} utterance randomly. The utterances before n^{th} are regarded as the input. If the remained dialogue utterances are less than 5, the example is discarded. *Last* chooses the last utterance of each dialogue for prediction. Moreover, *Sec* splits the dialogue into several sections and then picks the last utterance of each section. "Sec" is segmented based on the rule which requires each section to contain at least 8 utterances with at least 5 words and 3 significant utterances whose tf-idf value is superior to the threshold.

Figure 3 shows the result conducted on justice dataset[4]. It proves that meaningful utterance benefits the performance. Specifically, *Last* leads to the worst result on both R-1 and R-L due to the universal utterance at the end of a dialogue. We consider that *Random* prevents semantic information deficiency through selecting crucial utterances occasionally compared with *Sec* which achieves the best performance.

[4]The performance on AMI dataset shows a similar pattern. We only show the visualized result on the justice dataset due to the paper length limitation.

6 Conclusion

This work investigates the problem of unsupervised dialogue summarization. we propose a novel unsupervised strategy RepSum, which roots from the hypothetical foundation that a superior summary approximates a replacement of the original dialogue, and they are roughly equivalent for completing auxiliary tasks. RepSum is employed on both extractive and abstractive-based models via a self-supervision from two auxiliary tasks. Comprehensive experiments on various datasets show the effectiveness of the proposed mechanism compared to the other unsupervised baselines.

7 Acknowledgments

We sincerely thank Wei Liu, Yu Duan, and Jie Zhou for the helpful discussions. This research was supported by the National Key Research And Development Program of China (2018YFC0830200; 2018YFC0830206; 2020YFC0832505)

References

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. SEQˆ3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 673–681, Minneapolis, Minnesota. Association for Computational Linguistics.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online. Association for Computational Linguistics.

Trung H Bui, Matthew Frampton, John Dowding, and Stanley Peters. 2009. Extracting decisions from multi-party dialogue using directed graphical models and semantic similarity. In *Proceedings of the SIGDIAL 2009 Conference*, pages 235–243.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

Eric Chu and Peter Liu. 2019. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*, pages 1223–1232.

Thibault Févry and Jason Phang. 2018. Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422, Brussels, Belgium. Association for Computational Linguistics.

Prakhar Ganesh and Saket Dingliwal. 2019. Abstractive summarization of spoken and written conversation. *arXiv preprint arXiv:1902.01615*.

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

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 8.

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019. Automatic dialogue summary generation for customer service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1957–1965.

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.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Chris J Maddison, Daniel Tarlow, and Tom Minka. 2014. A* sampling. In *Conference and Workshop on Neural Information Processing Systems*, pages 3086–3094.

Inderjeet Mani. 2001. Summarization evaluation: An overview.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 319–328, Austin, Texas. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

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

Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. 2014. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 45–53, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.

Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. 2017. Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 12–21, Valencia, Spain. Association for Computational Linguistics.

Raphael Schumann, Lili Mou, Yao Lu, Olga Vechtomova, and Katja Markert. 2020. Discrete optimization for unsupervised sentence summarization with word-level extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5032–5042, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Lu Wang and Claire Cardie. 2013. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405, Sofia, Bulgaria. Association for Computational Linguistics.

Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. 2019. BottleSum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. 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 3752–3761, Hong Kong, China. Association for Computational Linguistics.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

Jiawei Zhou and Alexander Rush. 2019. Simple unsupervised summarization by contextual matching. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5101–5106, Florence, Italy. Association for Computational Linguistics.